

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра компьютерных технологий

«Допущен к защите»
Заведующий кафедрой _____

(Ф.И.О., ученая степень, звание)
_____ «____» _____ 20__ г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка модели алгоритмов для СЭС
Проект в Конечном виде

Специальность _____

Выполнил (а) Шимонаков Ю.В. БЭГ-12-3
(Фамилия и инициалы) группа

Научный руководитель Турганбаев Э.С., к.ф.н.
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Бекмичев А.У., к.э.н., доцент
(Фамилия и инициалы, ученая степень, звание)
_____ «04» 06 2016 г.
(подпись)

по безопасности жизнедеятельности:

Триходов Н.Г., д.х.н., проф.
(Фамилия и инициалы, ученая степень, звание)
_____ «02» 06 2016 г.
(подпись)

по применению вычислительной техники:

Турганбаев Э.С., к.ф.н.
(Фамилия и инициалы, ученая степень, звание)
_____ «____» _____ 20__ г.
(подпись)

Нормоконтролер: Турганбаев Э.С., к.ф.н.
(Фамилия и инициалы, ученая степень, звание)
_____ «____» _____ 20__ г.
(подпись)

Рецензент: Бабуханов Ф.Ф.
(подпись)
Турганбаев Э.С., к.ф.н.
(Фамилия и инициалы, ученая степень, звание)
_____ «____» _____ 20__ г.
(подпись)

Алматы 2016 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Агроинженерных и информационных технологий
Специальность Высокоскоростная техника и информационные системы
Кафедра коллективных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Исмаилов Юрий Викторович
(фамилия, имя, отчество)

Тема проекта Разработка модели административной сети
«Работа» Инженерная

утверждена приказом ректора № 21 от «10» марта 2016 г.

Срок сдачи законченной работы «__» _____ 20__ г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

Разработка инженерной модели
Административной сети

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

Анализ предметной области
Сбор требований заказчика к проекту
Исследование существующей структуры
Разработка логической модели и архитектуры
Разработка серверной части
Итерация
Технико-экономическое обоснование
Выявление успешности

Аннотация

Данная дипломная работа посвящена разработке коммерческого продукта – административной панели для справочно-правовой системы «Правовой консультант». Продукт является веб–приложением и спроектирован с учетом современных тенденций в дизайне и программировании. Приложение состоит из двух частей: клиентской части и серверной. Пояснительная записка к дипломной работе состоит из пяти глав.

В первой главе подробно рассматриваются требования заказчика к продукту на основе существующей панели администратора. Во второй главе рассказывается о языковых и программных средствах, используемых в разработке. В третьей главе представлен концепт и подробное описание проекта. В четвертой главе отображена разработка функционала и интерфейса приложения. Пятая и шестая главы посвящены экономической составляющей и анализу условий труда.

Annotation

This thesis is devoted to the development of a commercial product - the administrative panel for reference and legal system "Правовой консультант". The product is a web-based application and is designed in accordance with modern trends in design and programming. The application consists of two parts: a client-side and server. Explanatory note to the thesis work consists of five chapters.

In the first chapter of the report focuses on customer product requirements on the basis of the existing admin panel. The second chapter describes the language and software tools used in the design. The third chapter presents a concept and a detailed description of the project. In the fourth chapter and displayed functional application interface. The fifth and sixth chapters are devoted to the economic component and analysis of working conditions.

Андатпа

Бұл дипломдық жұмыс «Құқықтық кеңесші» анықтамалық-құқықтық жүйесі үшін әкімшілік панелі коммерциялық өнімін жобалауға арналған. Өнім веб-үстеме болып табылады және дизайн мен бағдарлаудың замануи беталыстарын ескере отырып жобаланған.

Үстеме екі бөлімнен тұрады: тұтынушылық бөлім және серверлік. Дипломдық жұмыстың түсіндірме жазбасы алты тараудан тұрады. Бірінші тарауда тапсырыс берушінің өнімге қоятын талаптары берілген әкімшілік панелі негізінде егжей-тегжейлі қарастырылған. Екінші тарауда жобада қолданылатын тілдері және бағдарламалық қамтамасыз ету құралдары туралы берілген. Үшінші тарауда жобаның концепті және толық сипаттамасы берілген. Төртінші тарауда үстемнің функционалы мен интерфейс көрсетілген. Бесінші және алтыншы тараулар экономикалық құрылымы мен еңбек жағдайларын талдамалауға арналған.

Содержание

Введение.....	8
Глава 1 Описание требований к системе СПС «Правовой консультант».....	10
1.1 СПС «Правовой консультант» описание системы	10
1.2 Основная проблема существующей административной панели	11
1.3 Параметры для обработки документа.....	12
1.4 Требование к панели администратора на первом этапе.....	13
1.5 Требование к панели администратора на втором этапе.....	15
1.6 Параметры для форматирования документа.....	19
1.7 Требования к гиперссылкам	20
1.8 Требование к рубрикатору	20
1.9 Дополнительные сервисы.....	21
1.10 Компиляция	31
1.11. Заключительные положения	33
Глава2 Средства Разработки	34
2.1 Клиентская часть.....	34
2.2 Серверная часть.....	38
2.3 База данных.....	39
2.4 Инструментальная среда разработки	39
Глава 3. Описательная часть проекта.....	40
3.1 Концепция проекта	40
3.2 Информационная архитектура.....	40
3.3 Дизайн и пользовательский интерфейс	40
3.4 Структура проекта	41
Глава 4 Реализация проекта	42
4.1 Программная структура и код	42
4.2 Реализация разделов	47
Глава 5 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА	53
5.1 Описание работы и обоснование необходимости	53
5.2 Трудовые ресурсы, используемые в работе	53

5.3 Оборудование и программное обеспечение, используемое в работе.....	53
5.4 Расчет стоимости разработки ПО.....	54
5.5 Сроки реализации проекта.....	54
5.6 Расчет фонда оплаты труда.....	55
5.7 Расчет затрат по социальному налогу.....	61
5.8 Расчет амортизационных отчислений.....	62
5.9 Расчет затрат на электроэнергию.....	62
5.10 Расчет накладных расходов.....	63
5.11 Цена реализации.....	64
Глава 6. Безопасность жизнедеятельности.....	66
6.1 Анализ условий рабочего помещения.....	66
6.2 Организация и оборудование рабочего места.....	67
6.3 Организация труда и отдыха при работе за компьютером.....	69
6.4 Освещение.....	70
6.5 Расчет освещенности помещения.....	70
6.6 Расчет уровня шума.....	72
Заключение.....	74
Список литературы.....	75
Приложение А.....	76

Введение

СПС «Правовой консультант» содержит нормативные правовые акты казахстанского законодательства по гражданскому, уголовному, административному, международному и хозяйственному праву. Разработчиками сделан основной акцент на нормативную и разъяснительную документацию по финансам, налогообложению, таможенному делу, валютному контролю, по рынку ценных бумаг, пенсионному обеспечению, бухгалтерскому учету (КСБУ, МСФО), страхованию, инвестициям, природопользованию, внешнеэкономической деятельности, а также другой полезной информации в сфере предпринимательской деятельности. Для наполнения СПС «Правовой консультант» используются все официальные источники информации. Объем документов около **30 000**.

СПС «Правовой консультант» это:

1. Полнотекстовая база данных по законодательству Республики Казахстан с полным гипертекстом, юридической обработкой документов и справкой по пользованию.

2. Комментарии к законодательству, разъяснения, справочная информация, словари, глоссарии.

3. Семь видов поиска, личные настройки, распечатка текстов документов, сетевая версия.

4. Еженедельное автоматическое обновление через Интернет, по e-mail и ежемесячно на CD.

5. Демонстрационная версия, техническая поддержка, обучение пользователей.

6. Минимальные технические требования к компьютеру и сети, удобный интерфейс.

7. Возможность отслеживать новые и измененные нормативные правовые акты и документы.

Данный дипломный проект направлен на удовлетворение требований по обновлению и мониторингу системы юристами компании и представляет собой удобное и функциональное веб-приложение.

С учетом того, что у компании, поставляющей данный продукт, имеется локальная панель администратора имеющая множество недостатков и не отвечающая стратегическим целям компании, тема моей дипломной работы актуальна для этой компании.

Также учитывая тот факт, что при разработке данного приложения использовались передовые технологии программирования, неизученные исполнителем ранее, можно сделать вывод об актуальности проекта для исполнителя, то есть меня.

В настоящее время быстрыми темпами идет глобализация и внедрение всех приложений в сеть интернет, для обеспечения постоянного и обширного доступа к последним. На основании этого можно сделать вывод об актуальности приложения в глобальном смысле этого слова.

Целями данного дипломного проекта являются:

1. Анализ требований заказчика, необходимых для разработки и внедрения приложения

2. Разработка и утверждение Технического задания

3. Разработка непосредственно приложения

4. Внедрение

Нельзя не отметить особенности панели администратора:

1. Удобный и интуитивно понятный интерфейс

2. Быстрота работы

3. Широкий спектр возможностей для юристов

Глава 1 Описание требований к системе СПС «Правовой консультант»

1.1 СПС «Правовой консультант» описание системы

Как уже ранее говорилось: «Правовой консультант» - профессиональная справочная правовая система, продукт компании «InterConsult Info».

Содержит нормативные правовые акты законодательства Республики Казахстан и инструментарий для работы с ними.

Представлена на рынке локальной и онлайн версиями.

Ниже продемонстрирована главная страница СПС «Правовой консультант»:

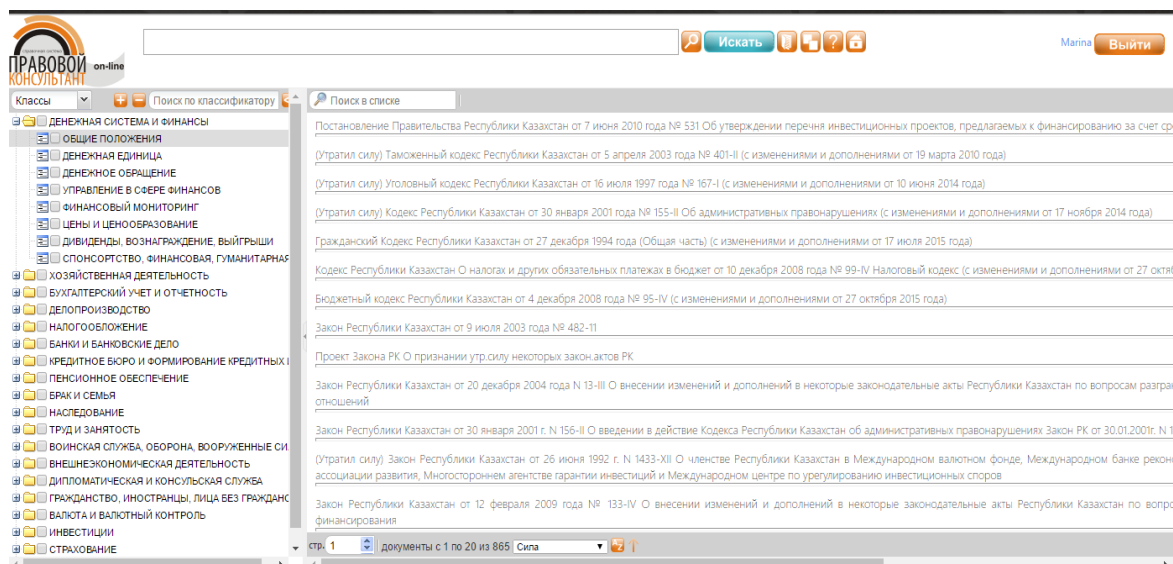


Рисунок 1 – Главная страница СПС «Правовой консультант»

Из рисунка видно, что приложение представляет собой удобный справочник с поиском необходимых документов по классификаторам и ключевым словам.

На Рисунке 2 можно увидеть довольно больше количество функции для работы с документами в правой части экрана.

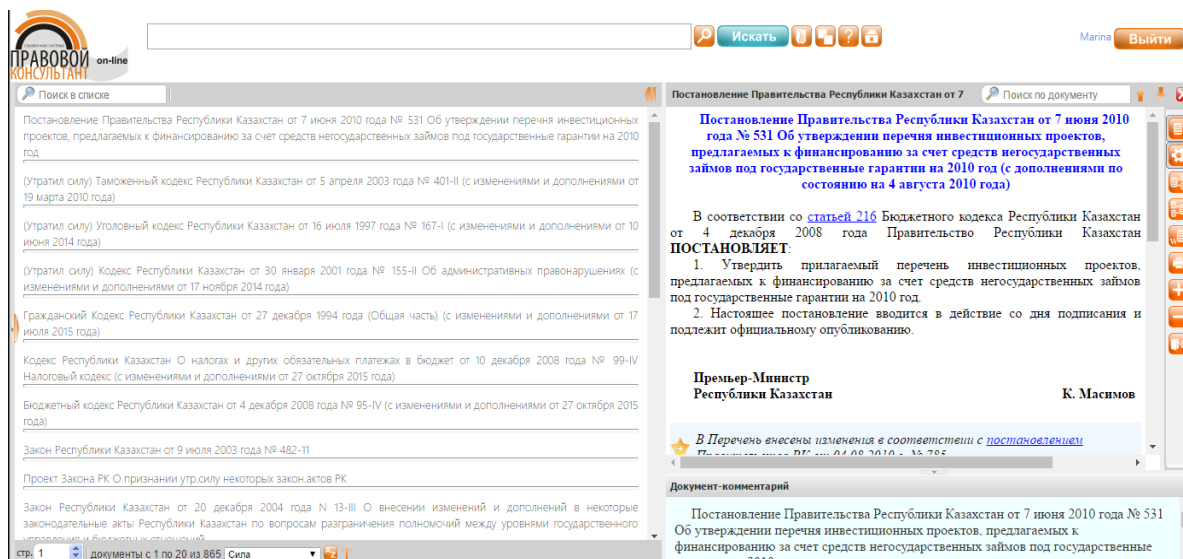


Рисунок 2 – Выбор конкретного документа и функционал для работы с ним.

Данная система довольно конкурентоспособная за счёт ее небольшой стоимости, компактности и функциональности.

1.2 Основная проблема существующей административной панели

Законодательная база постоянно обновляется, следовательно, возникает необходимость в постоянном обновлении системы с целью поддержки спроса и конкурентоспособности. Как же выглядело обновление системы посредством панели администратора до внедрения моего продукта?

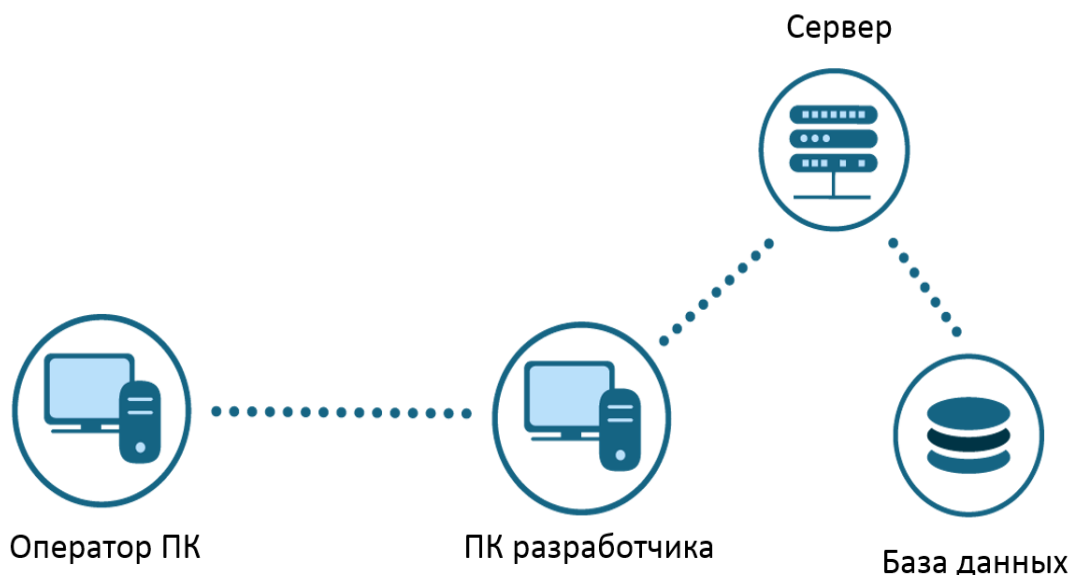


Рисунок 3 – Последовательная цепь обновления СПС «Правовой консультант»

Юристы по мере обновления действующего законодательства РК обязаны вносить новые документы в систему посредством локальной административной панели. Оператор ПК производит определенную работу по добавлению, удалению и другим функциям с программой, после чего формируется скрипт, хранящий в себе данные добавления. Затем данный скрипт передается на компьютер разработчиков системы, что занимает много времени. Разработчик в свою очередь запускает скрипт на сервере и происходит обновление базы данных.

Учитывая тот факт, что компания вскоре хочет полностью интегрировать свой продукт в сеть интернет и отказаться от локальной версии, было решено разработать веб-приложение для работы оператора с сервером напрямую.

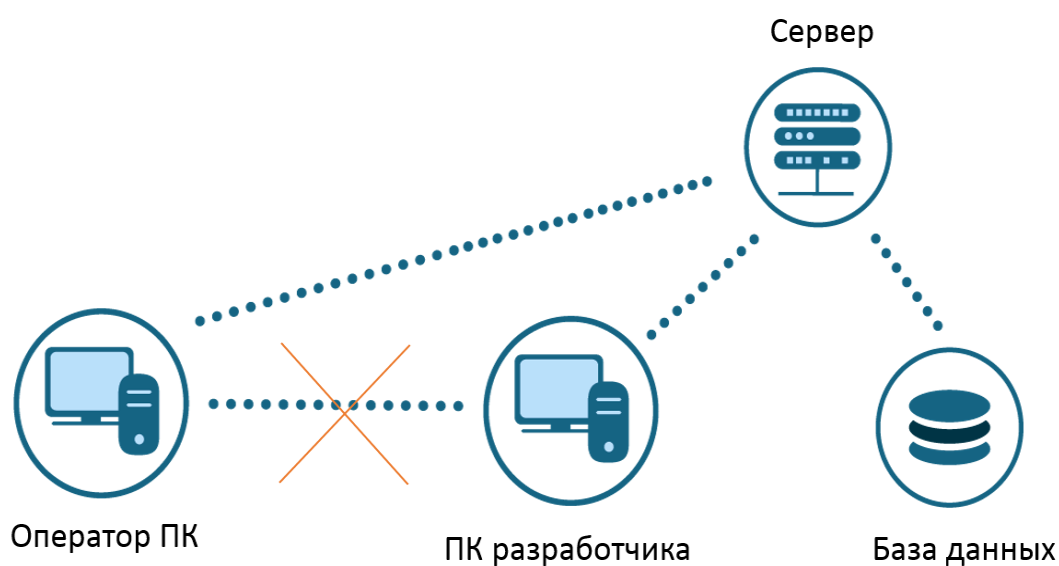


Рисунок 4 – Последовательная цепь обновления после внедрения новой административной панели

1.3 Параметры для обработки документа

1. Вставляем текст в базу (текст может содержать картинки, формулы, таблицы). Перед тем, как добавить документ кодификатор проверяет его наличие в базе. Текст копируется из разных источников и имеет разное форматирование, лишние теги и ссылки.

1.1. Автоматически убрать все «вражеские» ссылки и черный цвет, жирность, чтобы кодификатор имел их ввиду

1.2. Сохранить форматирование первоисточника (жирность, курсив, выравнивание текста)

1.3. Шаблон для любого текста: автоматом текст ставить шрифтом - Times New Roman, размер -12, но оставлять возможность для иного

форматирования. Отступы и абзацы во всех документах должны быть одинаковы. См. п.45

1.4. При копировании максимально сохранить качество рисунков, таблиц. Если рисунок сразу будет вставать в текст – идеальный вариант.

2. Если рисунок может быть загружен только через прикрепления файла, то автоматизировать данный процесс до максимума. Кодификатор не должен сам присваивать ему название, оно должно формироваться автоматически, брав за основу определенный код: например номер id документа + еще что-то, но каждый номер должен быть уникальный (по принципу как ставятся якоря), с учетом, что этот номер будет виден клиенту. Нужен приличный номер.

1.4 Требование к панели администратора на первом этапе

В таком случае библиотека рисунков должна быть представлена в виде таблицы:

1. id атачмента
2. id документа с атачментом
3. присвоенное имя атачменту, под которым он будет сохраняться для клиента (см. ниже принцип сохранения для клиента)
4. описание атачмента, которое кодификатор дает только для себя для собственного удобства на случай поиска или дальнейшего редактирования
5. размер атачмента
6. дата загрузки атачмента, если будут изменения, то их фиксировать, также: когда, кем, на каком основании внесены
7. - имя кодификатора, загрузившего файл

Панель по функционалу:

1. Выберите файл (одновременно можно выбрать любое кол-во файлов)
2. Закачайте
3. Фильтры для поиска
4. сохранить/изменить/заменить/удалить/чистка

Функции по прикрепленным файлам:

1. Сохранить – то есть выбрать и добавить в библиотеку атачмент
2. Изменить – изменяется только его описание;
3. Заменить – значит поменять внутреннее содержание атачмента, но оставить его исходный id
4. Удалить – удаление атачмента
5. Чистка – построить список атачментов, которые не привязаны ни к одному документу, для их удаления или прикрепления

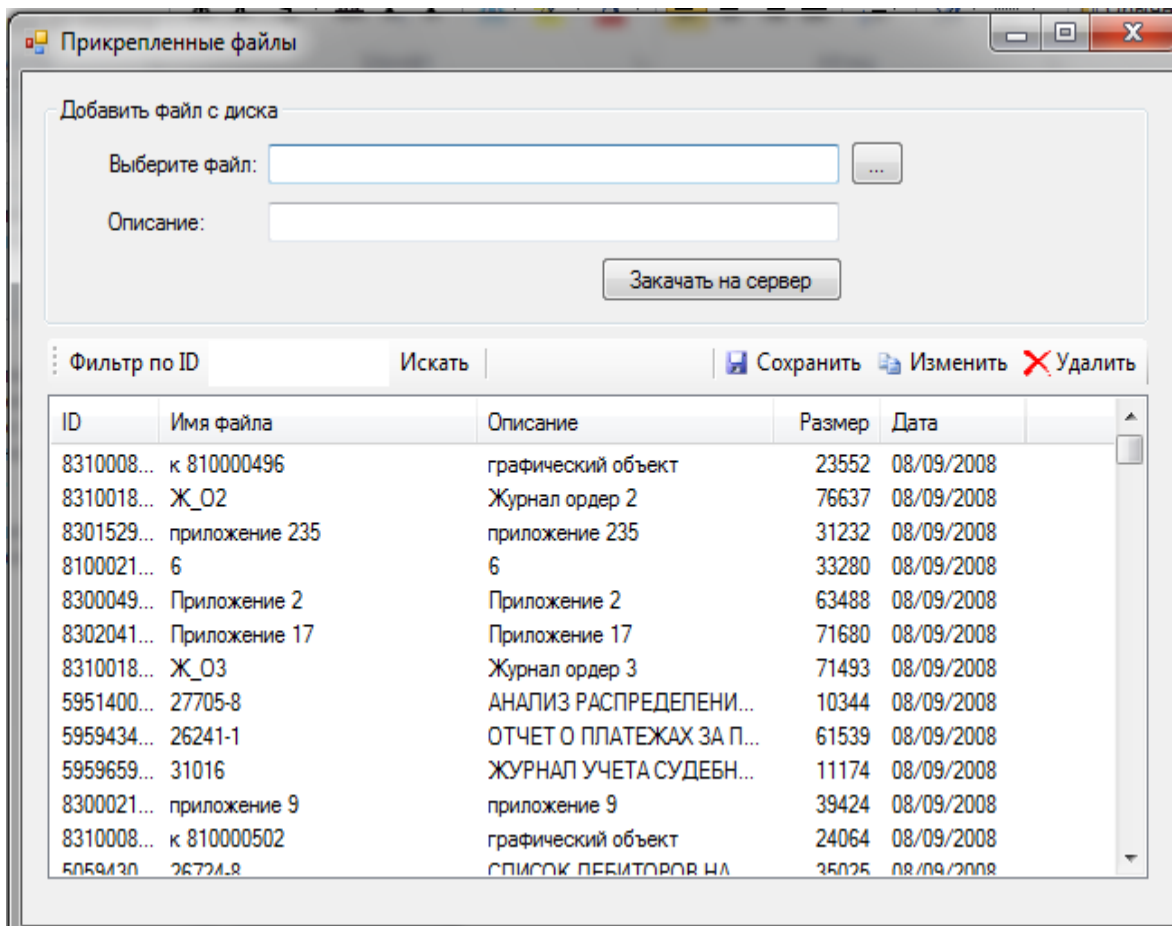


Рисунок 5 - Библиотека файлов в настоящее время

Работа с рисунком – как это видит клиент:

Клиент открывает текст с рисунком, просматривает и может сохранить:

Весь документ с рисунками в формате html, в таком случае выходит окно с выбором директории для сохранения и возможностью ему присвоить любое название, примерно так:

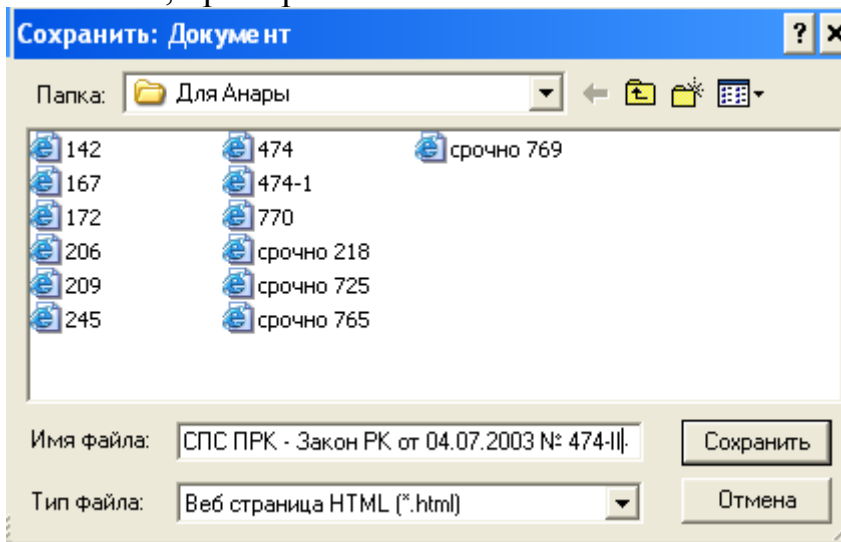


Рисунок 6 – Сохранение документа

Предложение с сохранением под таким названием должно формироваться автоматически по следующим принципу:

СПС ПРК – Тип документа (если их несколько берется тот, что по списку первый или приоритетный, по ходу работы определимся) – дата издания – номер

Если какого-то атрибута нет, он просто пропускается.

Таким образом, сохраняется документ в формате html + папка, в которой все прикрепленные изображения под номерами, которые присвоены автоматически п.4 этапа 1.1. Картинки должны сохраняться по такой же очередности, как они встречаются в тексте.

Весь документ с рисунками в формате word.

Механизм тот же, только возможность открыть или сохранить. Имя присваивается также, только рисунки должны быть сразу в тексте.

Рисунки прикрепляются без ограничения в объеме. Бывают случаи, что один рисунок побит на несколько составных частей и все части нужно посадить как один файл. Учесть возможность такого объединения. Также предусмотреть статистику по прикрепленным файлам, чтобы контролировать максимальный объем выгрузки для локальной версии. В действующей панели администратора такое ограничение есть - 4 мб. В созданной - сделать уведомление, если объём превышает 10 мб.

1.5 Требование к панели администратора на втором этапе

В документе может быть несколько дат, номеров (номера бывают в виде букв, латинских букв и т.д.)

Типы дат документа:

1. дата издания
2. дата опубликования
3. дата регистрации в МинЮсте

Действие: редактировать (всплывающий календарь), то есть возможность изменить или удалить дату. Дата издания и опубликования может быть несколько или не быть вообще.

Типы номеров:

1. номер документа при издании
2. номер документа при регистрации в МинЮсте

Действие: редактировать, то есть возможность изменить или удалить номер. По каждому типу может быть несколько номеров или не быть вообще

Сейчас это представлено так:

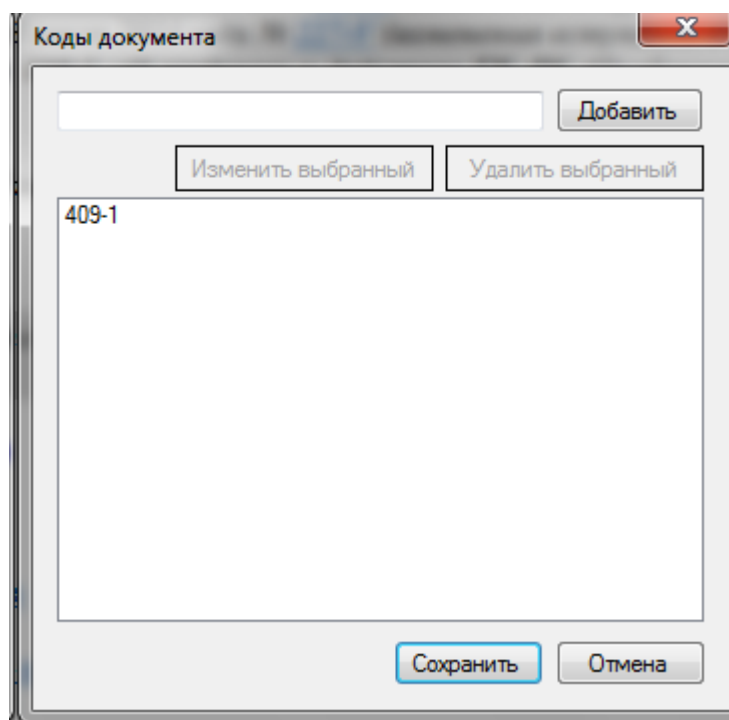


Рисунок 7 – Окно кода документа

Название в трех вариациях:

А) Полное официальное наименование (учесть чистку невидимых разрывов – случай 1с) – 1шт.

Б) Сокращенное наименование – 1шт.

В) Народное название - наименование согласно сленгу, названию в народе, облегченных сокращений для поиска (ПДД, МРП, ставки КПН и тд.). Перечислением, без запятой. Виден только в панели.

Г) Название для колонтитулов – 1 шт (при печати в колонтитулах должен отображаться статус документа – утративший силу или не введен в действие, в действующем ничего не указывать).

Поиск организован по всем категориям (как в СПС, так и в панели).

Действие: возможность редактировать любое название.

Приоритет согласно сформированному списку.

Список редактируется только администратором, с полной свободой менять значения приоритета. При смене значения приоритета, он автоматически меняются во всех документах, где ранее был проставлен. Любой кодификатор может отправить запрос на добавление нового приоритета, если база не найдет совпадений, то администратор увидит запрос в отдельном поле и добавит нужный приоритет.

Сейчас это представлено так:

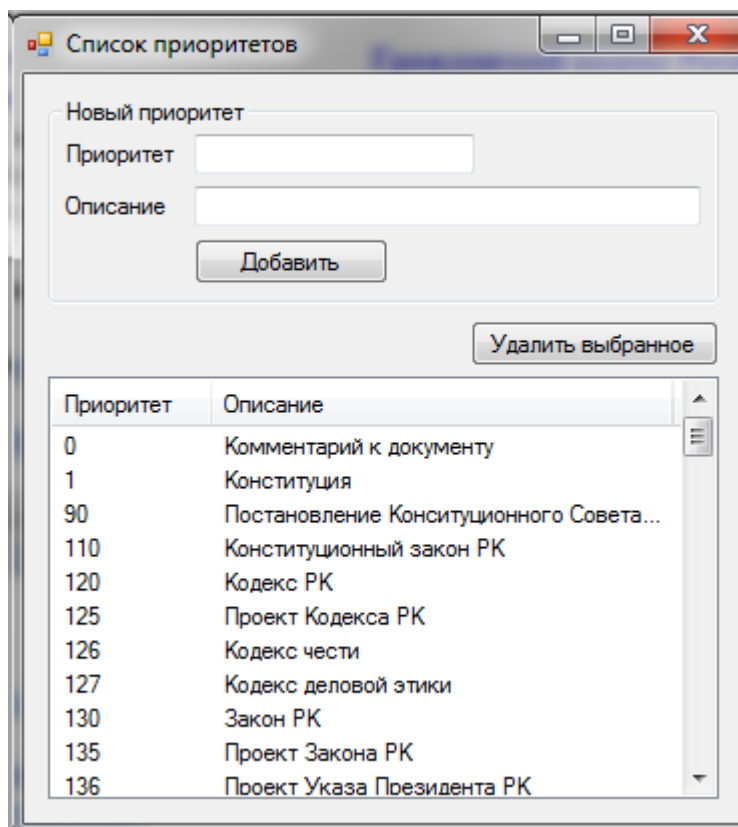


Рисунок 8 – Список приоритетов

При конвертации в 2004 году часть документов села с приоритетами, которых нет в действующих списках, например, 32767. Нужна возможность вывести документы, в которых приоритеты не из списка, чтобы почистить базу.

Варианты статусов к документу:

1. Действующий – виден всем
2. Утративший силу – виден всем
3. Не введен в действие – виден всем
4. В стадии обработки – виден всем
5. Старая редакция – виден всем
6. Отменен – виден всем
7. Тестовый документ – виден только кодификаторам
8. Внутренний документ – виден только кодификаторам

Действие: возможность менять статусы к документам. При смене статуса, к названию документа автоматически, в начале, в скобках добавляется соответствующая надпись (не применять к действующим).

Например: (Утратил силу) Закон РК О страховании...

Черновик – один из дополнительных статусов к документу, означающих, что данный документ редактируется и не может быть отображен в рабочей версии СПС (ни при каких условиях). Все кодификаторы имеют доступ к такому документу. Если идет ссылка на этот документ с другого текста, должно выходит окно, примерно такой текст, еще согласуем.

«Запрашиваемый вами документ имеется в СПС «Правовой консультант», но находится на стадии обработки. Пожалуйста, повторите ваш запрос после текущего обновления».

Не документ – данный документ присутствует в базе и виден только кодификатору, участвует в поиске только панели.

Комментарий – один из дополнительных статусов к документу, означающий, что данный документ не участвует в общем поиске у клиента. Его можно открыть только через функцию – справка к документу – из основного документа. У кодификатора он ищется как любой другой документ. Предназначен для документов-комментариев (справка к документу). Как только кодификатор выбирает данный статус, программа предлагает ему список документов, с которыми ему надо связать этот комментарий. Кодификатор должен видеть связь, где главный документ, где комментарий к нему. Если удаляется главный документ, база предлагает сразу удалить его комментарий.

Создаем к документу комментарий

Создается как самостоятельный документ. Всегда следует за основным документом. Подробнее см. п.3.6. Текст комментария можно редактировать как любой другой документ. Обязательные поля для заполнения: название (согласно разделу, а) и б) пп.3.2, не является документов. Связь между ними (номер id друг друга можно посмотреть в любом из документов).

Сохраняем и удаляем документ

Панели сохраняет документ при заполнении всех обязательных полей и после проверки на наличие идентичного документа в базе (администратор может снимать запрет). Создание комментария обязательно. Панели предупреждает, если сохраненный документ имеет ошибки в теге. И сразу заносит этот документ в отдельный раздел (на случай возникших проблем с 1с). После устранения ошибок и повторного сохранения, документ сохраняется в обычном режиме.

Обязательные атрибуты, без которых невозможно сохранение:

- *Название в двух вариациях:*
- *Полное официальное наименование (учесть чистку невидимых разрывов – случай 1с)*
- *Сокращенное наименование*
- *Приоритет согласно сформированному списку.*

Варианты сохранения:

- сохранить;
- сохранить и отобразить в мониторинге обновлений – по умолчанию

К каждому сохраненному документу прилагается «окно-справка», в котором идет полный отчет о том, когда, кто создал документ, когда и какие

(при раскрытии функции) были внесены изменения. И пометки кодификатора относительно документа, примечания.

После сохранения выводится список документов (при необходимости можно закрыть), в которых встречается название, дата (точное совпадение) и номер (точное совпадение) созданного документа, кодификатор может выбирать документы, на которых автоматически появится ссылка на новый документ (можно выбрать весь список). Это при условии, если не побьется форматирование.

Удаляем документ.

Документ удалить может любой кодификатор, но после подтверждения администратором. В окне пагеди у администратора должно появиться предупреждение, о том. Что кодификатор N желает удалить документ id. Администратор проверяет документ и после подтверждения удаления – документ удаляется из базы.

1.6 Параметры для форматирования документа

Форматируем согласно требованиям

1. Автоматическое редактирование согласно п.1 главы 1.
2. В больших текстах с главами присутствует оглавление, ссылки должны проставляться на разделы автоматически (как в Word, виды заголовков)
3. Библиотека символов и формул, возможность ставить сноски и прикреплять текст (графические объекты, архивированные файлы).
4. Заносим данные из текста в раздел «словари», «определения». При выделении кнопкой «слово – определение» автоматически предлагать занести его в словарь терминов или определений, если там таковое есть, предложить сравнить источник и/или обновить значение.
5. Возможность выделить текст в комментарий с любой возможностью его редактирования (удалить текст, изменить, дополнить или отменить):
 - a. *Пп. 3 изменен в соответствии с Законом РК от 01.03.2011 г. № [414-IV](#)*
6. Отображение таблиц и графических объектов. Сложные формулы: химические, математические, физические значения. Фотографии, подписи, гербы. Библиотека уже прикрепленных файлов – должны быть видны рисунки. Обтекание рисунков. К каждому документу показывать объем и список прикрепленных файлов, с возможностью их редактирования (внести изменения, удалить, заменить).
7. Предварительный просмотр текста с отключением комментариев.
8. Функцию найти/заменить/заменить все (с учетом регистра и окончания)
9. Сохранять старую редакцию текста, как приложение к основному документу. Старая редакция не участвует в поиске, не прописывается и на нее не ссылаются другие документы, ссылки из текста старой редакции – рабочие.

Думаю старую редакцию документа нужно пересохранять под новым id, чтобы извне никто на него не ссылался. Список старых редакций по дате отображается в комментарии к документу.

10. Прикрепляем файлы, например pdf, word

Общие функции:

Набор функций для работы с текстом и таблицами, приближенные к Word представлен на Рисунке 9.

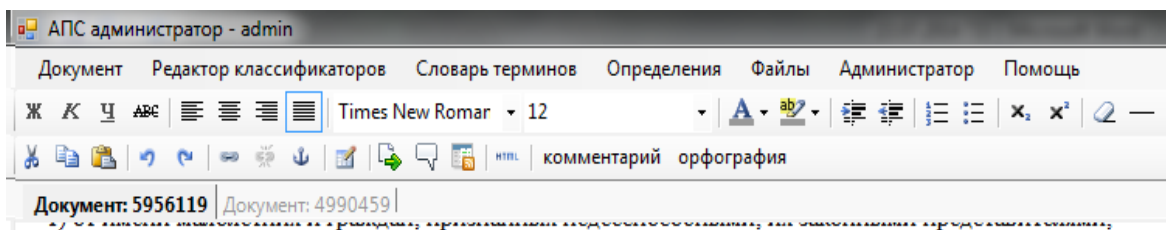


Рисунок 9 – Набор функции для работы с текстом и таблицами

1.7 Требования к гиперссылкам

Ссылки ставятся:

- на сам документ
- на фрагмент самого документа
- на иной документ
- на фрагмент иного документа
- мультиссылки (на несколько документов, на их начало и/или часть текста)

После удаления документа, замены статуса (утратил силу) возможность пересмотреть адреса ссылок, автоматизировать процесс замены адреса ссылок на новый (действующий) документ.

Видеть, какой документ ссылается на часть текста, утратившего силу и возможность подкорректировать значение (на случай, если статья утратила силу, перенаправить ссылку). То есть если стоит якорь – то видеть с какого/на какие документы он ведет.

Видеть корреспондентов и респондентов в панели.

Если текст, на который ставится ссылка, повторяется по тексту, автоматически найти ее и проставить ссылки. Учитывать регистр и окончание. Но только с разрешением такого действия кодификатором.

Если документ утратил силу, предлагать список действующих документов, которые на этот утративший ссылались, для того, чтобы перенаправить их новый документ.

1.8 Требование к рубрикатору

Разделы Классы/типы/источники/по ситуации, а также возможность самим создавать новые разделы, редактировать их (объединять, разъединять) и удалять

Кодификаторы могут выбирать классы/типы/источники только из созданных разделов, если раздел не существует, они могут создать раздел-черновик. Администратор подтверждает его создание или отклоняет.

Разделы классификатора можно менять местами, переименовывать, выделять из общего потока подразделы, создавать новые, подклассы, объединять и тд (доступ имеет только администратор). Открывать при этом документы из разделов не нужно. Документы следуют за перемещенным классом.

Все манипуляции по изменению принадлежности документа к классу/типу/источнику можно производить, не открывая сам документ (доступ кодификаторам и администратору). Просто выстраивается список документов и галочками (как реализован почтовик mail.ru) отмечаются документы и нужное для них действие (переместить, удалить, пометить). Помеченные документы, по мимо своих классах лежат еще в общем классе, который на заметку у кодификатора, как документ, требующий исправления.

Предлагать кодификатору наиболее подходящие (подобные) классы (типы, источники). Это должно браться из смысла полного наименования.

Если раздел удален, выводить список документов, которые остались без этого класса. Пока список не будет отработан весь, раздел удалить нельзя.

1.9 Дополнительные сервисы

Электронная книга учета

1. для каждого пользователя, администратор имеет доступ ко всем книгам
2. должен быть раздел, где фиксируется автоматически весь объем работы кодификатора, список документов и манипуляции с ним
3. кодификатор может создавать заметки, приписки к документу или своей работе
4. фильтровать свои документы по статусы: в процессе обработки, черновик, не введен в действие, по дате и т.д.
5. памятки, напоминки, уведомления кодификатору от администратора

Удаленный сотрудник

1. минимизировать настройку, идентичные возможности
2. администратор может урезать возможный функционал
3. администратор может зайти удалено
4. база показывает какие кодификаторы работают, удалено или стационарно, сколько времени подключены и их активность

Добавление новых пользователей и их права (доступно только администратору)

Разделить по возможностям и правам доступа: администратор, пользователь, удаленный сотрудник, ученик

Серверные возможности

1. запрет на одновременное редактирование одного документа несколькими кодификаторами
2. предупреждение на вновь созданный документ о наличии такого же документа в базе
3. возможность одновременно ставить ссылки на один и тот же документ (документ может быть открыт/закрыт)
4. администратор может закрыть доступ кодификатору ко всей базе (отключить)

Библиотека прикрепленных файлов

Решить вопрос перехода от старого принципа прикрепления файлов к новому, как конвертировать все прикрепленные файлы без потери данных и связей. Для этого нужна возможность просматривать, изменять, удалять, дополнять прикрепленные файлы, заменять адрес или добавлять эти файлы к нескольким документам.

В процессе работы создавать библиотеку часто встречаемых рисунков (кодификаторы могут сами ее дополнять)

Уведомление о превышении объема обновления 10Mb

1. выгрузка производится по решению администратора, лимита на объем нет
2. подсчет предполагаемого объема обновления
3. предварительный просмотр списка документов

Чистка панели (можно организовать как фильтры в поиске)

Выводится список документов по требуемым запросам, с указанием (подсветка) текста

1. имеются вражеские ссылки
2. черновики
3. не введенные в действие
4. некорректное название (отсутствует полное/сокращенное название, нет даты)
5. не корректные ссылки (42.0.26.78)
6. на прикрепленный файлы, если основной документ удален/не существует
7. возможны другие варианты

Словарь терминов и определений

1. заносить слова в списки словарей, редактировать их, поиск, фильтры
2. предупреждать, если источник определения утратил силу

3. не допускать повторов, за исключением, если источники разные, можно сделать несколько вариантов определений к одному слову
4. если слово занесено в словарь, кодификатор должен видеть это по тексту
5. каждое действие в словаре фиксируется в истории изменений, администратор может просмотреть историю
6. пополнять списки из Интернета (википедия)

Документы на казахском языке (английском языке)

1. к основному документу идет привязанный документ на казахском языке. Переключение одной кнопкой
2. актуальность документа и все прописки (атрибуты), ссылки делаются как к основному документу
3. максимально автоматизировать процесс

Утративший силу текст

Сервис, который делает утративший силу текст серым цветом и зачеркнутым. Кодификатор определяет границы, делаем комментарий-обоснование. При желании клиент может «отключить» текст, который утратил силу, в таком случае в СПС на местах текста, утратившего силу, появляется - ИКЛЮЧЕН (на этом слове ссылка на комментарий-обоснование).

Если будет сделан сервис – старая редакция, то нет необходимости в п.11.

Но надо решить, как переконвертировать все прежние документы.

Комментарии к тексту – текст, который сам набирает кодификатор внутри документа. Данный текст носит информативный характер, он может быть изменен, удален.

1. при выделении текста в комментарий автоматом делать его Times New Roman 12 курсив.
2. возможность ставить комментарий в таблице или в любой другой части документа
3. редактирование комментария: изменить, удалить, дополнить.

Комментарий к документу

1. в тексте комментария автоматически должны сохраняться списки старой редакции
2. соответствующий комментарий идет к документу на русском и казахском языке, также переключатся: каз/рус

Старая редакция

Перед внесением изменений, кодификатор может внести копию всей редакции в раздел: Старая редакция, то есть в архив к данному документу. Кол-во старых редакций не ограничено. При занесении в старую редакцию

пишется обоснование и весь список документов доступен в комментарии к документу см. п.6.9. Кодификатор может редактировать текст старой редакции, удалить.

Например: Старая редакция - [Кодекс](#) РК «О налогах и других обязательных платежах в бюджет» от 10 декабря 2008 года № 99-IV (Налоговый кодекс) (с изменениями и дополнениями по состоянию от 10 июля 2012 года, действовавший до 1 января 2013 года).

Это же функция может использоваться как резервное сохранение документа.

Помощь-справка к панели администратора

1. разработать правила, описание кнопок и требования по работе с панели
2. обновление, изменение правил производит администратор

Автоматическое сохранение и резервное восстановление при экстренном выключении компьютера или другом сбое

Автоматическое сохранение резервной копии документа – приблизить как в Word, на случай отключения компьютера после входа в панель предложить загрузить последний редактируемый документ с внесенными, но не сохраненными изменениями. Резервное восстановление – возможность сделать откат базы с выбором времени. Полагаю, это будет 1 день.

Закрытие панели, на случай если панель закрылась случайно

1. при экстренном закрытии/отключении сохранять последнюю версию и предлагать ее открыть, статус к такому документу – черновик
2. при закрытии панели с несколькими открытыми документами предложить совершить действия с ними: сохранить/не сохранить изменения (если таковые есть)

Приоритеты документов

1. список создается, редактируется администратором
2. кодификатор может отправить администратору на подтверждение новый, измененный приоритет

Тестовая панель с полным набором функционала для стажеров

Функция html

видеть исходный html текста и организовать там поиск, найти/заменить, сохранить как.

On-line контроль для администратора

1. видеть в реальном времени работу каждого кодификатора
2. выполненный объем за определенный промежуток времени

Статичный текст

1. название документа выделяется в «статичный текст» и не зависимо от прокрутки, остаётся в шапке
2. оглавление проставляется автоматически, идет в статичный текст по решению кодификатора (если не большой объем, надо подумать)
3. при желании кодификатор может отключить режим «статичный текст»

Поиск в админке

1. по классам в рубрикаторе
2. по ID номеру
3. по названию (включает все виды названий) и добавить возможности поиска как в он-лайне
4. поиск по дате,
5. по номеру
6. в результатах поиска галочками отметить документы, необходимые открыть или документы, в которых надо поменять классы или статус (например: приказ и 10 регламентов, поставить статус – утратил силу)
7. в результате поиска сразу видеть его статус (черновик, не введен и тд)
8. в остальном поиск с учетом окон из старой админ Рисунке 10.

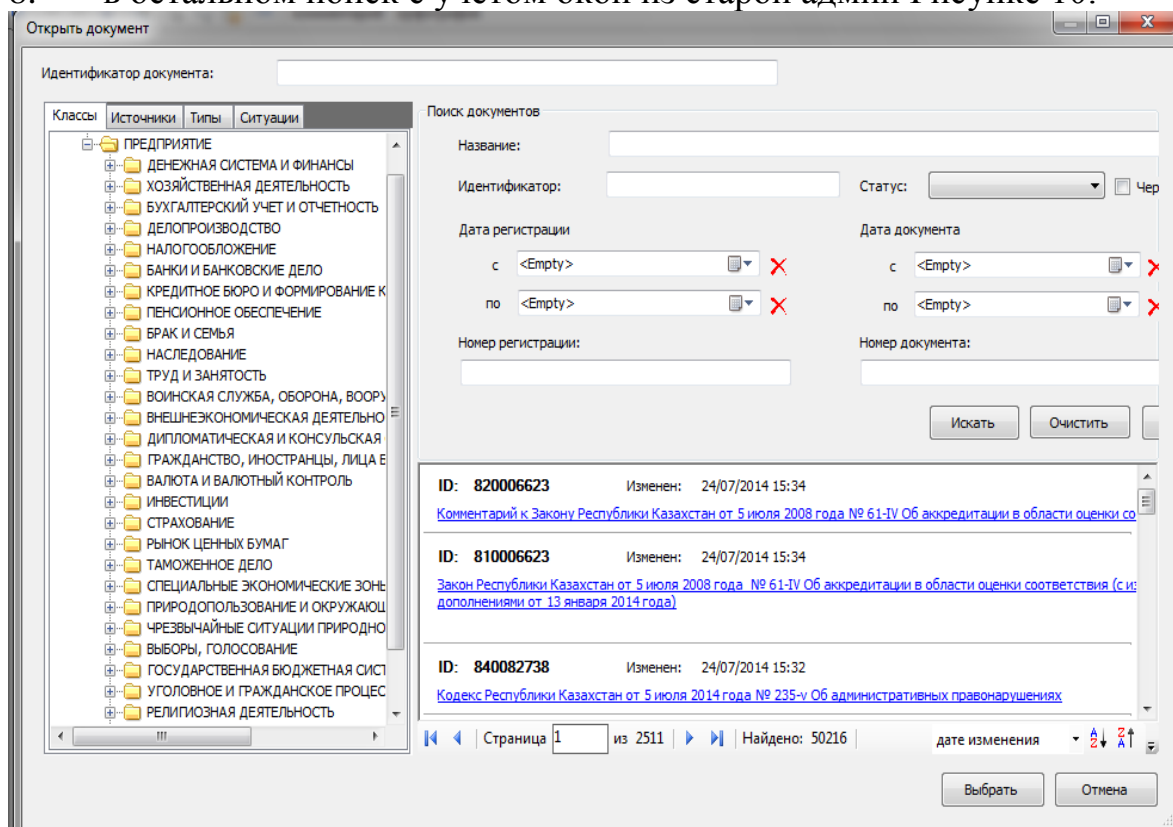


Рисунок 10 – Расширенный поиск

Обновление версии панели

1. выводить данные о программе: версия
2. предупреждать о новом релизе, автоматически предлагать обновить версию

Кнопка удаления общего форматирования, с возможностью убрать форматирование только на выделенном фрагменте

Функция отменить/вернуть

1. не ограниченное количество кликов;
2. вернуть версию до работы определенного кодификатора или показать работу определенного кодификатора с этим документом (доступно только администратору).

Диаграммы и колонки

Возможность скопировать их в панель из Word

Возможности классификатора

1. возможность самостоятельно создавать продукты из классификатора (доступно только администратору). Делать сборку из нескольких классов, и только эта подборка видна клиенту
2. создавать, редактировать, удалять, любые манипуляции с классификатором (доступно только администратору)
3. при манипуляции с классификатором, в документах автоматически идет сохранение новые классов, если класс должен быть расформирован или удален, то сначала нужно список документов, который состоит в этом классе, перенаправит в новые разделы

Предусмотреть функцию для ценовой оценки документа на будущие продажи через смс и др.

Ревизия документов по статусам

желательно выводить список по заданным значениями. Например, показать все не введенные в силу документы, или документы, в которые вносятся изменения с такой-то даты

Счетчик

1. по предварительному количеству документов в выгрузке
2. НПА
3. комментариев к документу
4. справочных материалов

Принятые сокращения, аббревиатура (можно учесть в п.15 главы 5)

1. для государственных органов
2. для НПА
3. для физических значений

Уведомления для кодификаторов

1. предупреждение создается кодификатором или администратором и видно всем. Все предупреждения весят списком
2. предупреждение о том, что скоро в документ будут внесены изменения или что документ будет изменен с момента опубликования этих изменений, за 3 дня до наступления данной даты
3. предупреждение о том, что документ имеет временное ограничение по действию и уведомлять кодификатора за 1 день до предполагаемого события.
4. предупреждение о том, что документ начинает действовать с определенной даты, за 1 день до наступления данной даты
5. такие документы должны по своему статусу выстраиваться по поиску, для их контроля
6. предупреждение выходит у кодификатора, который создал или последний работал с этим документом и у администратора
7. после внесения соответствующих изменений кодификатор делает пометку в предупреждении. После проверки администратором данное предупреждение снимается.

Задание

1. администратор может делать пометки и предупреждения, давать задание в электронном журнале кодификатора
2. кодификаторы могут делать заметки (но не изменять) в электронных журналах других кодификаторов

Образцы стилей текста

Так как в базе уже реализован процесс расстановки ссылок (якоря), то надо учесть их перенос в новую админку. Возможно, оставить тот же принцип.

При применении любого стиля удаление якорей не допускается.

1. Якорь – знак, не видимый для пользователя, ставится вручную кодификатором для создания на него ссылки из части другого текста.
2. При печати и отображение в Word сохраняются все стили
3. Можно реализовать другой вариант, но с учетом сохранения уже имеющего, чтобы не побились ссылки

Автоматически весь текст:

Шрифт - Times New Roman, размер - 12, абзац - 0,5, междустрочный - одинарный; интервал перед – 0, интервал после – 0, гиперссылки - адрес удалять и помечать жирным черным цветом, абзац – 0,5;

оставлять как в оригинальном тексте:

жирность, цвет текста (чтобы видеть официальный текст и комменты вражеские), выравнивание текста

Пример:

1. Законодательство Республики Казахстан об обязательном страховании гражданско-правовой ответственности аудиторских организаций основывается на **Конституции** Республики Казахстан и состоит из **Гражданского кодекса** Республики Казахстан, настоящего Закона и иных нормативных правовых актов Республики Казахстан.

2. Если международным договором, ратифицированным Республикой Казахстан, установлены иные правила, чем те, которые содержатся в настоящем Законе, то применяются правила международного договора.

Все заголовки к тексту:

1. жирным шрифтом, выравнивание по центру, цвет синий (синий 255: оттенок 160, контраст 240, яркость 120, красный 0, зеленый 0), интервал перед – 12пт, интервал после – 6пт, абзац - 0

2. Исключение: название текста (самый первый заголовок) – интервал перед -0, интервал после – 30пт

Пример:

Приказ Министра юстиции Республики Казахстан от 16 ноября 2012 года № 379 О внесении изменения в приказ Министра юстиции Республики Казахстан от 27 мая 2010 года № 171 «Об утверждении Инструкции по проведению юридической экспертизы договоров займа, заключаемых под гарантии Республики Казахстан»

Все комментарии к тексту:

Курсив, абзац 0,5, вся строка с текстом комментария выделяется в светло-серый цвет, выравнивание по левому краю.

Пример:

Владельцам автомобильного транспорта возмещаются расходы за использование транспорта в случаях, предусмотренных настоящей статьей, а также причиненный ущерб за счет государственного бюджета в порядке, установленном гражданским законодательством Республики Казахстан.

Закон дополнен ст.5-1 в соответствии с Законом РК от 03.07.2013 г. № [121-V](#)

Все ссылки, проставляемые в панели

Не жирный синий цвет с подчеркиванием

Пример:

Законодательным актом о статусе столицы Республики Казахстан могут быть установлены особенности перевозки пассажиров и багажа автомобильным транспортом в столице.

Заголовки боковые (в основном встречаются справа, на всякий случай разрешить выбор справа-слева)

Равнение по правому краю, жирный синий цвет.

Пример:

**Приложение
к приказу Председателя
Агентства Республики Казахстан
по делам строительства и
жилищно-коммунального хозяйства
от 10 декабря 2012 г. № 603**

Информационное окно о статусе (ставится между основным названием документа и началом текста носит информативный характер)

Жирный черный текст, курсив, абзац 0,5, равнение по левому краю, интервал перед – 0, после -30 пт

Пример:

**Приказ и.о. Министра труда и социальной защиты населения
Республики Казахстан от 27 сентября 2007 года № 225-п**

Утратил силу в соответствии с приказом Министра внутренних дел РК от 30 марта 2011 года № 135

В соответствии с пунктом 8 статьи 34 Закона Республики Казахстан «О миграции населения» **ПРИКАЗЫВАЮ:**

Рассмотрим другой вариант, например, ставить его в рамки.

При печати в колонтитулах должен отображаться статус документа – утративший силу или не введен в действие, в действующем ничего не указывать.

Все названия статей (обычно всегда слева)

1. Применяются на название статей (на конкретно выделенный текст)

2. Жирный синий цвет, абзац – 0,5, интервал перед – 6пт, после – 0 (если сложно это сделать, то кодификатор может сам делать отступы)

Пример:

Статья 2. Законодательство Республики Казахстан об автомобильном транспорте

Законодательство Республики Казахстан об автомобильном транспорте основывается...

Статья 3. Сфера действия настоящего Закона

Действие настоящего Закона распространяется на всех физических и юридических лиц, осуществляющих в соответствии с законодательством Республики Казахстан деятельность в сфере автомобильного транспорта.

Статья 4. Основные принципы осуществления деятельности в сфере автомобильного транспорта

Работы и услуги в сфере автомобильного транспорта ...

Слова-определения (не требует создания стиля, по возможности)

Синий цвет

Пример:

Статья 1. Основные понятия, используемые в настоящем Законе

В настоящем Законе используются следующие основные понятия:

1) **система экстренного вызова при авариях и катастрофах (далее - система экстренного вызова)** - государственная ...

1-1) **автобус** - автомобильное транспортное средство, предназначенное

...

2) **автовокзал** - комплекс сооружений, предназначенный ...

2-1) **услуги автовокзалов, автостанций и пунктов обслуживания пассажиров** - деятельность, осуществляемая ...

Статус текста – утративший силу, не введенный в действие текст надо подумать, какие возможны варианты? Желателен сервис для отключения его отображения, печать/не печать, старая редакция

На сегодня текст утративший силу выглядит так:

~~8) разрабатывают программу развития автомобильных перевозок пассажиров и багажа, в том числе внутрирайонных и сельских перевозок;~~

Пп.8 п.2 ст.14 исключен в соответствии с Законом РК от 03.07.2013 г. № [124-V](#)

Предлагаю отнести этот текст к статусу комментария к тексту (п.38), чтобы при отключении функции «комментарий», отключалось и отображение текста, утратившего силу. Помечать текст утративший силу так:

~~8) разрабатывают программу развития автомобильных перевозок пассажиров и багажа, в том числе внутрирайонных и сельских перевозок;~~

Описание причины так:

Пп.8 п.2 ст.14 исключен в соответствии с Законом РК от 03.07.2013 г. № [124-V](#)

Текст, не введенный в действие, на сегодня никак не отображается, только в начале документа в информационном окне (п.41) пишется о вероятных изменениях в документе. Иногда кодификатор оформляет эти изменения в комментарии так:

Для целей настоящего подпункта:

С 1 января 2014 год абзац 2 части 4 пп.1 п.3 ст.244-1 действует в новой редакции в соответствии с Законом РК от 04.07.2013 г. № [132-V](#):

«регулярным рейсом признается рейс, выполняемый согласно расписанию, установленному и опубликованному авиакомпанией в порядке, определяемом законодательством Республики Казахстан об использовании воздушного пространства Республики Казахстан и деятельности авиации»

Регулярным рейсом признается рейс, выполняемый согласно расписанию, утвержденному в порядке, установленном [законодательством](#) Республики Казахстан об использовании воздушного пространства Республики Казахстан и деятельности авиации;

Содержание (оглавление)

Равнение по левому краю, цвет синий, интервал применяется ко всему списку содержания: перед – 30пт, после – 30пт, абзац – 0

[Закон Республики Казахстан от 4 июля 2003 года № 476-III Об автомобильном транспорте](#)

[\(с изменениями и дополнениями от 4 июля 2013 года\)](#)

[Глава 1. Общие положения \(ст.ст. 1 - 9\)](#)

[Глава 2. Государственное регулирование в сфере автомобильного транспорта \(ст.ст. 10 - 19\)](#)

[Глава 2-1. Осуществление государственного контроля в области автомобильного транспорта \(ст.ст. 19-1 - 19-11\)](#)

[Глава 3. Автомобильные перевозки пассажиров и багажа \(ст.ст. 20 - 29\)](#)

[Глава 4. Автомобильные перевозки грузов и почты \(ст.ст. 30 - 41\)](#)

[Глава 5. Международные автомобильные перевозки \(ст.ст. 42 - 45\)](#)

[Глава 6. Ответственность на автомобильном транспорте \(ст.ст. 46 - 55\)](#)

Настоящий Закон регулирует отношения, возникающие между перевозчиками, пассажирами, грузоотправителями, грузополучателями, другими физическими и юридическими лицами в сфере автомобильного транспорта.

1.10 Компиляция

Периодичность

Автоматически, каждый день с 19-20ч. админка должна производить выгрузку документов для on-line и локальной версии.

Мониторинг

После выгрузки должен формироваться мониторинг по:

1. новым документам
2. измененным документам
3. не введенным в действие документам
4. утратившим силу документам

Список документов должен сворачиваться по месяцам, годам.

Данный мониторинг должен быть на главной странице локальной и он-лайн версии. Так же доступен для его просмотра из администраторской части.

Обновления История просмотра

■ - новые ■ - измененные
■ - утратившие силу ■ - не введенные в действие

Июль 2014

1	24 июля 2014	1	51	7	5
2	22 июля 2014	14	179	41	3
3	18 июля 2014	19	83	31	1
4	17 июля 2014	15	25	11	1
5	16 июля 2014	14	73	45	2
6	15 июля 2014	7	61	17	4
7	11 июля 2014	8	119	22	2
8	09 июля 2014	10	66	9	0
9	08 июля 2014	0	50	29	0
10	07 июля 2014	0	8	3	1
11	04 июля 2014	2	24	3	2
12	03 июля 2014	18	154	39	2
13	01 июля 2014	3	347	71	0

Июнь 2014

1	27 июня 2014	1	286	57	0
2	26 июня 2014	12	417	71	2

Рисунок 11 – Мониторинг обновлений

Статистика

Данные об общем количестве документов, формируются автоматически. Статистика должна быть на главной странице локальной и он-лайн версии. Так же доступна для просмотра из админской части.

Статистика включает в себя:

1. количество нормативно-правовых актов
2. количество справочных материалов и комментариев
3. количество приложений (это прикрепленные файлы)

Количество нормативно-правовых актов: 28062	Количество справочных материалов и приложений: 34472	Обновление от: 24.07.2014
---	--	---------------------------

Рисунок 7 - Статистика

Список клиентов из биллинг системы

Админка должна быть связана с биллинг системой, для того, чтобы выгруженные файлы с обновлением могли установиться только у

зарегистрированных пользователей. Скачать их может любой пользователь, установить – только зарегистрированный.

Автоматическая загрузка на ресурс

После выгрузки (компиляции) сформированный файл (дельта) должен быть автоматически выложен на ресурс для скачивания клиентами. Локальная версия СПС сама делает запрос о наличии доступных обновлений и всплывающим окном уведомляет об этом пользователя. По выбору пользователя или внутренним настройкам клиент обновляет базу

Обратная связь

Биллинг-система должна получить отзыв о скачивании файла, для контроля со стороны компании о произведенных обновлениях.

Вынужденная компиляция

По производственной необходимости, администратор может произвести выгрузку вручную.

1.11. Заключительные положения

Конвертация

Все документы должны пройти конвертацию с устранением ошибок:

1. присвоением head и body
2. если нет короткого наименования документа, дать его по имеющемуся полному наименованию
3. вывести список с документами, в которых наименование прописано заглавными буквами
4. исправить разрывы в наименовании (полном и коротком)
5. вывести список документов с указанием мест на битые ссылки

Полное внедрение казахского языка

Для основного текста и прикрепленным файлам

Работа админ в экстренном режиме

На случай отсутствия интернета возможность временной работы по локальной сети

Глава2 Средства Разработки

2.1 Клиентская часть

Как уже сообщалось ранее разработка административной панели будет представлять из себя разработку клиентской части и серверной части.

Учитывая тот факт, что мы имеем дело с веб-приложением, нельзя не применить всему известную верстку на HTML, стилизацию на CSS и JavaScript.

HTML

HTML является языком программирования, позволяет разрабатывать веб-сайты. Ему сравнительно легко учиться, во-первых, в применении, во-вторых, существует множество документации в свободном доступе. Он постоянно подвергается пересмотру и эволюции для удовлетворения потребностей, и требований растущей интернет-аудитории под руководством W3C, в организации, занимающейся проектированием и сохранением языка.

HTML - язык разметки гипертекста.

Гипертекстом является метод, с помощью которого вы передвигаетесь по веб-страницам нажимая на специальные текстовые гиперссылки, которые переносят вас на следующую страницу.

HTML является языком, так как он имеет кодовые слова и синтаксис, как и любой другой язык.

Пример кода:

```
<!-- Right(Notification and Searchbox -->
<ul class="nav navbar-nav navbar-right">
  <li>
    <!-- Notification -->
    <div class="notification-box">
      <ul class="list-inline m-b-0">
        <li>
          <a href="javascript:void(0);" class="right-bar-toggle">
            <i class="zmdi zmdi-notifications-none"></i>
          </a>
          <div class="noti-dot">
            <span class="dot"></span>
            <span class="pulse"></span>
          </div>
        </li>
      </ul>
    </div>
  </li>
</ul>
```

Рисунок 12 – Пример списка на HTML

Что такое CSS?

CSS - каскадные таблицы стилей:

1. CSS описывает, как HTML элементы, которые будут отображаться на экране, бумаги, или в других средствах массовой информации

2. CSS экономит много работы. Он может контролировать расположение нескольких веб-страниц все сразу

3. Внешние таблицы стилей хранятся в файлах CSS

Зачем использовать CSS?

1. CSS используется для определения стилей для ваших веб-страниц, в том числе дизайн, верстка и вариаций дисплея для различных устройств и размеров экрана.

2. CSS Решенные большая проблема

3. HTML никогда не предназначался для содержать теги для форматирования веб-страницы!

HTML был создан для описания содержимого веб-страницы, как:

<H1> Это заголовок </ h1>

<P> Это параграф. </ P

Когда теги, как и цветовые атрибуты были добавлены в спецификации HTML 3.2, он начал кошмар для веб-разработчиков. Развитие крупных веб-сайтов, где шрифты и информация о цвете были добавлены к каждой отдельной странице, стал длительный и дорогостоящий процесс.

Чтобы решить эту проблему, Консорциум World Wide Web (W3C) создал CSS. Еще немного фактов:

1. CSS удалены стиль форматирования со страницы HTML!

2. CSS экономит много работы!

3. Определения стилей обычно сохраняются во внешних файлах .css.

4. С помощью внешнего файла таблицы стилей, вы можете изменить внешний вид всего веб-сайта, изменив только один файл.

JavaScript

JavaScript является языком программирования высокого уровня, динамичный, нетипизированным и интерпретируемый. Имеет стандартизацию в спецификации языка ECMAScript. Наряду с HTML и CSS, он является одной из трех основные технологии производства Wide Web контента; большинство сайтов используют его, и он поддерживается всеми современными веб-браузерами без плагинов. JavaScript является прототипом в основе с функциями первого класса, что делает его язык фундаментальным и поддерживающим объектно-ориентированный, императивные и функциональные стили программирования. Он имеет API для работы с текстовыми, массивами, датами и регулярными выражениями, но не включает в себя ввод / вывод, такие как создание сетей, систем хранения данных, или графических объектов, опираясь на них по факту принимающей среды, в которую он встроен.

Хотя есть сильные внешние сходства между Javascript и Java, в том числе название языка, синтаксис и соответствующие стандартные библиотеки – это два различных языка, отличающиеся по своей конструкции. JavaScript находится под влиянием языков программирования, таких как Self и Scheme.

JavaScript используется не только в средах, связанных с веб-интерфейсом, но и таких как PDF-документы, специфические браузеры и

настольные виджеты. Новые и быстрее JavaScript виртуальных машин (VM) и платформы, построенные на них, также увеличили популярность JavaScript для веб-приложений на стороне сервера. На стороне клиента, JavaScript традиционно реализован как интерпретируемый язык, но более поздние браузеры выполняют его в момент компиляции. Он также используется в разработке игр, создании настольных и мобильных приложений, а также сетевого программирования на стороне сервера, такие как Node.js.

Функционал клиентской части администраторской панели было решено реализовать на популярном JS фреймворке AngularJS.

AngularJS

Фреймворк в значительной степени позволяет расширить возможности HTML путем интеграции в него множества директив, обрабатывающих данные. Придает динамичности HTML коду. Помимо всего здесь наследованы все программные возможности нативного JavaScript. Используется преимущественно в разработке SPA приложений.

Почему именно AngularJS?

Большое сообщество

Комьюнити включает в себя в качестве участников постоянного развития множество команд или просто тех, кто хочет внести свой вклад в усовершенствование фреймворка с открытым исходным кодом. Постоянно проводятся различные конференции и мастер классы, существует множество документации и книг в интернете.

Динамический и упрощенный стиль кода

В простом JS:

```
var students = [
  {name: 'Sara Connor', score:'97'},
  {name: 'John Connor', score:'60'},
  {...}
];

var studentsList = document.getElementById('students-list');

students.forEach(function(e){
  var studentEntry = document.createElement('li'),
      studentName = document.createElement('span'),
      studentScore = document.createElement('span');

  studentName.innerHTML = e.name;
  studentScore.innerHTML = e.score;

  studentEntry.appendChild(studentName);
  studentEntry.appendChild(studentScore);

  studentList.appendChild(studentEntry);
});
```

Рисунок 13 – Фрагмент работы с произвольным DOM-деревом
Этот же код в AngularJS:

```
<div ng-controller="studentsController as studentsCtrl">
  <div ng-repeat="student in studentsCtrl.students">

    Name: <span class="name">{{student.name}}</span>
    Title: <span class="title">{{student.title}}</span>

  </div>
</div>
```

Рисунок 14 – Фрагмент работы с этим же DOM-деревом на AngularJS

Использование директив

Нативный HTML расширяется с помощью директив, которые описывают поведение различных элементов DOM-дерева.

MVC

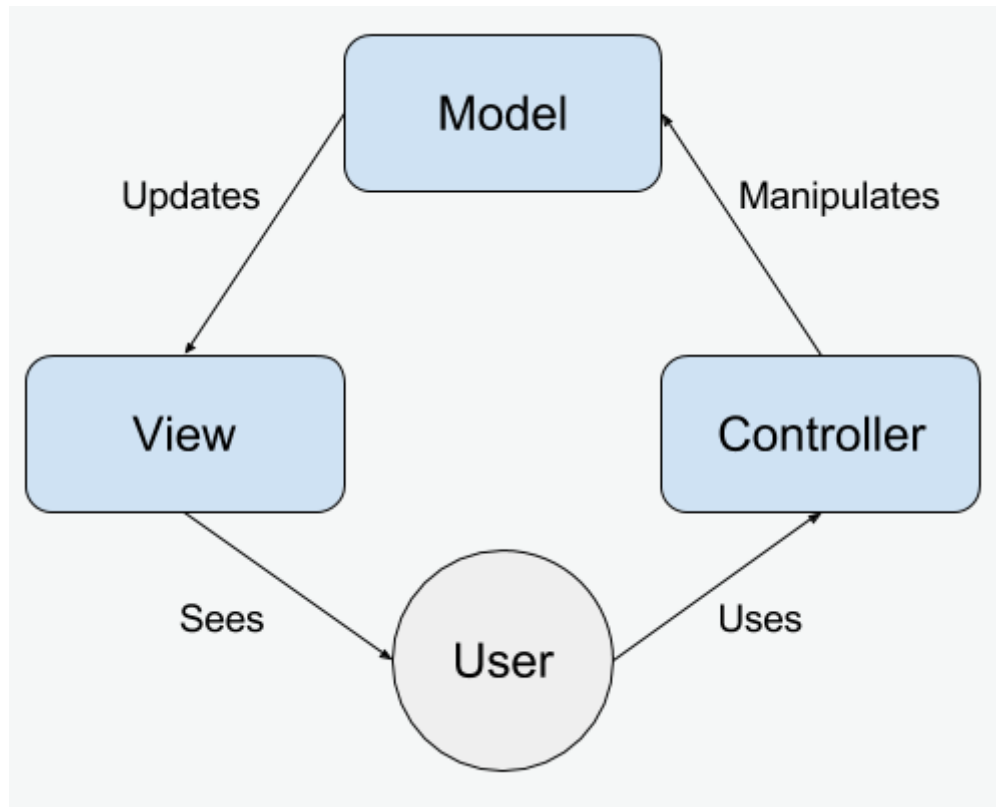


Рисунок 15 – Схема MVC

Позволяет отделить логику приложения от дизайна. Взаимодействует с сервером посредством службы \$http, при передаче объекта JavaScript автоматически преобразует его в JSON и наоборот.

Модульность

Приложение состоит из отдельных автономных или зависимых модулей.

Наличие готовых решений

Уже реализовано большое количество расширений, позволяющих упростить работу с Фреймворком и облегчить разработку веб-приложения. Возможность создавать проекты с различной структурой позволяют использовать фреймворк даже в разработке приложений на мобильные средства.

Двустороннее связывание данных

Все изменения, которые происходят в интерфейсе приложения, сразу же применяются к объектам приложения и наоборот. Фреймворк периодически отслеживает изменения в браузере, изменения в модели и события, чтобы обновлять нужные шаблоны.

Тестирование

В связи с модульностью фреймворка, автоматическое тестирование отдельных частей программы удобно и эффективно.

2.2 Серверная часть

Серверная часть приложения СПС «Правовой Консультант» построена на Java, поэтому целесообразно было бы использовать для администраторской панели существующий код, добавив в него сервлет обработки GET и POST запросов.

Java

Java является языком программирования и вычислительной платформой, первая выпущенная Sun Microsystems в 1995 году. Существует множество приложений и веб-сайтов, которые не будут работать, если вы не установили Java. Java является быстрой, безопасной и надежной. От портативных компьютеров до центров обработки данных, от игровых приставок до научных суперкомпьютеров, от мобильных телефонов до Интернета, Java везде.

Java является основой практически для каждого типа сетевого приложения и является мировым стандартом для разработки и внедрения встраиваемых и мобильных приложений, игр, веб-контентом и корпоративного программного обеспечения. Объединенная более чем 9 миллионов разработчиков по всему миру, Java позволяет эффективно разрабатывать, развертывать и использовать интересные приложения и сервисы.

От портативных компьютеров до центров обработки данных, от игровых приставок до научных суперкомпьютеров, от мобильных телефонов до Интернета, Java везде.

- 97% Enterprise Desktops Run Java
- 89% рабочих столов (или компьютеры) в США Run Java
- 9 миллионов Java разработчиков по всему миру
- № 1 выбор для разработчиков
- № 1 Платформа разработки
- 3 миллиарда мобильных телефонов Запуск Java
- 100% от Blu-Ray проигрыватели Корабль с Java

- 5 миллиардов Java-карты в использовании
- 125 миллионов ТВ устройства работают под управлением Java
- 5 Top 5 производителей оригинального оборудования судов Java

ME

2.3 База данных

Как в случае с Java система СПС «Правовой консультант» использовала СУБД MySQL.

MySQL

MySQL находится в открытом исходном коде, система предназначена для управления реляционными базами данных (СУБД). В июле 2013 года она была второй самой широко используемой СУБД в мире. Она названа в честь Майкла Widenius, в то время как "SQL" стоит как аббревиатура Язык структурированных запросов. Исходный код MySQL находится в свободном доступе в соответствии с условиями GNU General Public License, а также другими фирменными соглашениями. В настоящее время принадлежит корпорации Oracle. Для частного пользования доступны некоторые платные версии, предлагающие дополнительные функциональные возможности.

2.4 Инструментальная среда разработки

В ходе разработки административной панели использовался следующий софт:

1. IntelliJ IDEA – многопрофильная интегрированная среда разработки программного на многих языках программирование, в том числе Java, JavaScript, PHP, включающая в себя большое количество программных модулей и плагинов.

2. Sublime Text 3 – мощный текстовый редактор, поддерживающий синтаксис множества языков. Также снабжен большим количеством плагинов, которые облегчают редактирование кода. Удобен при верстке.

3. MySQL Workbench - продукт компании Oracle. Программа для работы с СУБД MySQL, написание запросов, функций и т.д. Снабжена функциями графического отображения БД.

Глава 3. Описательная часть проекта.

3.1 Концепция проекта

1. *Группы потенциальных посетителей*
 - a. Юристы компании
 - b. Кадровики
 - c. Управляющий персонал
2. *Маркетинговые задачи*
 - a. Уменьшение времени обновления системы с целью поддержания спроса на последнюю.
 - b. Постоянный мониторинг деятельности операторов с целью контроля эффективности работы последних.
 - c. Увеличение эффективности работы операторов.
3. *Цели размещение в сети Интернет*
 - a. Бесперебойный и открытый доступ в сети Интернет
 - b. Взаимодействие системы напрямую с сервером
4. *Программные задачи*
 - a. Простой и интуитивно понятный интерфейс
 - b. Удобный функционал для CRUD операций
 - c. Редактор документов с большим набором функций
 - d. Гибкий поиск
 - e. Система разграничения прав доступа
 - f. Мониторинг обновлений

На основе представленных требований необходимо разработать многопользовательское веб-приложение, отвечающее современным требованиям дизайна интерфейса и функционала. Акцент сделать на минимализме и интуитивно понятном интерфейсе, а также на удобном расширенном поиске необходимых документов по ключевым словам.

3.2 Информационная архитектура

1. Блок последних обновлений системы
2. Словарь терминов
3. Документы, акты, постановления
4. Классификаторы документов
5. Блок информации о пользователях

3.3 Дизайн и пользовательский интерфейс

Дизайн разработан на основе утвержденных Заказчиком модульных сеток типовых страниц.

Дизайн выполнен с учетом растягивающейся сетки.

На всех страницах сайта должны отображаться следующие элементы:

1. Логотип Компании;
2. Форма для поискового запроса;
3. Главное меню;
4. Внутренние баннеры (один блок, любое количество от 2 до 5);
5. Уведомления

3.4 Структура проекта

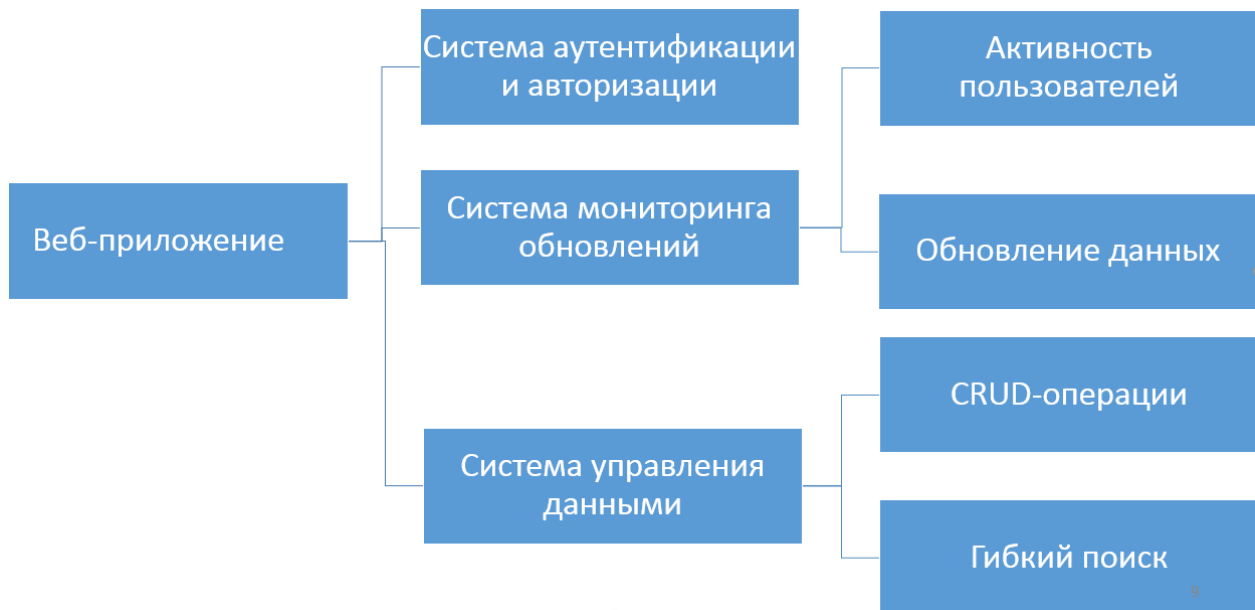


Рисунок 16 – Структура проекта

Глава 4 Реализация проекта

4.1 Программная структура и код

Разрабатываемая нами административная панель по сути есть одностраничное веб-приложение. Создаем необходимые директории с стандартов программирования.

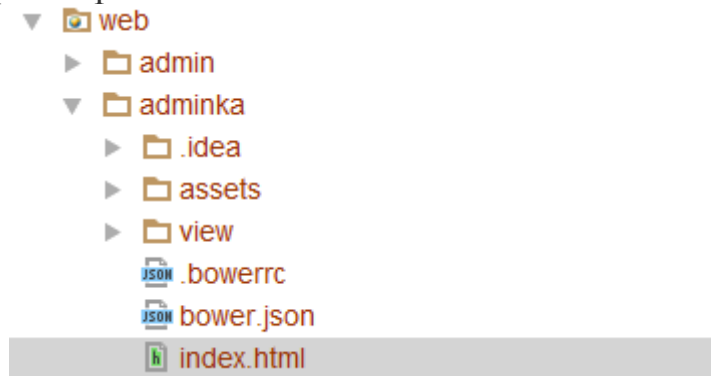


Рисунок 17 – корневая директория проекта

Из Рисунка 17 видно, что существует главный файл `index.html`, куда и подключаются все необходимые модули и библиотеки, здесь же инициализируется приложение:

```
<!-- Begin page -->  
<div ng-app="mainApp" id="wrapper">
```

Рисунок 18 – инициализация приложения

Информация о подключаемых библиотеках хранится в файле `bower.json`, создаваемым Bower – пакетным менеджером.

```

{
  "name": "adminka",
  "authors": [
    "shipovalovyuriy <shipovalovyuriy@gmail.com>"
  ],
  "description": "",
  "main": "",
  "license": "MIT",
  "homepage": "",
  "ignore": [
    "**/*.*",
    "node_modules",
    "bower_components",
    "assets/js",
    "test",
    "tests"
  ],
  "dependencies": {
    "angular-cookies": "~1.4.5",
    "angular-drag-and-drop-lists": "~1.3.0",
    "angular-route": "~1.4.4",
    "angular-bootstrap-nav-tree": "*",
    "angular-resource": "~1.4.4",
    "jquery": "~2.1.4",
    "angular-ui-grid": "~3.1.1"
  },
  "resolutions": {
    "angular": ">=1.0"
  }
}

```

Рисунок 19 – Все подключаемые модули и библиотеки

Рисунок 19 показывает, что в основном пакетным менеджером скачены готовые библиотеки AngularJS.

Подключаются они в главном файле index.html, как говорилось ранее:

```

<!-- jquery -->
<script src="assets/js/jquery.min.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script src="assets/js/detect.js"></script>
<script src="assets/js/fastclick.js"></script>
<script src="assets/js/jquery.blockUI.js"></script>
<script src="assets/js/waves.js"></script>
<script src="assets/js/jquery.nicescroll.js"></script>
<script src="assets/js/jquery.scrollTo.min.js"></script>
<!------->

```

Рисунок 20 – Пример подключения библиотек в главный файл
 В директории view хранятся все необходимые шаблоны каждого раздела:

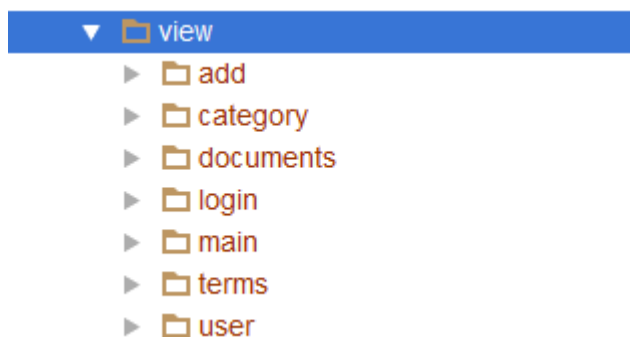


Рисунок 21 – Шаблоны разделов приложения
 В директории assets хранятся стили приложения и JavaScript файлы:

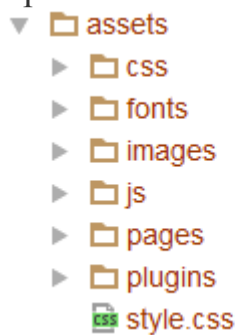


Рисунок 22 – Подключаемые библиотеки и стили
 Вся логика и реализация функционала в соответствии с требованиями Заказчика находятся в директории assets/js/app

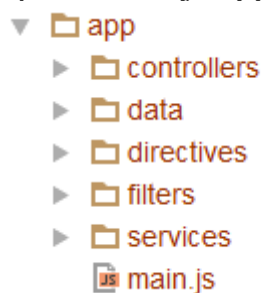


Рисунок 23 – Функционал приложения

Для обработки запросов пользователей и передачи в последствии этих запросов на сервер служат контроллеры. Каждый контроллер охватывает определенный раздел программы и работает автономно, независимо от других контроллеров, что делает систему гибкой для различного рода изменений. Пример реализации контроллера документов:

```
'use strict';

mainApp.controller("documentController", function($scope, $http, $location, Category, Document, uiGridConstants, $modal){

    $scope.classItems = [];
    $scope.docs = [];

    Category.getClass().then(function (data) {
        $scope.classificators = data;
    });

    $scope.treeHandler = function(branch){
        if (branch.children.length == 0){
            Document.getDocs(branch.id).then(function(data){
                $scope.docs = data;
                console.log(branch.id);
                console.log($scope.docs);
            });
        }
    }
}

$scope.getTree = function(id) {
    Category.getTree(id).then(function (data) {
        console.log(data);
        $scope.classItems = data;
    });
}

$scope.searchParams = {};
$scope.searchDocument = function(){
```

Рисунок 24– Программный код контроллера документов

Также имеются самописные директивы расширяющие возможности HTML языка:

```

'use strict';
mainApp.directive('edited', function () {

    return {

        link: function (scope, element, attr) {

            element.val(scope.classItem.name);

            element.on('keypress', function ($event) {
                if ($event.which == 13) {
                    scope.editCat(element.val(), scope.classItem);
                }
            });
        }
    };
});

mainApp.directive('added', function ($compile) {

    return {

        link: function ($scope, element, attr) {

            element.on('click', function () {
                var el = element.parent().find('ul:first');
                var y = angular.element($compile("<li><input ng-model='adding' type='text' /></li><spa
                var x = el.prepend(y);
            });
            //element.on('keypress',function($event){
            //
            // if( $event.which==13){
            //     console.log('asdfsadf');
            // }
            // })
        }
    }
}

```

Рисунок 25– Директива для редактирования и добавления классификатора

Сервисы и фильтры являются вспомогательными частями программы, там реализуются главные функции дабы не нагружать контроллеры.

4.2 Реализация разделов

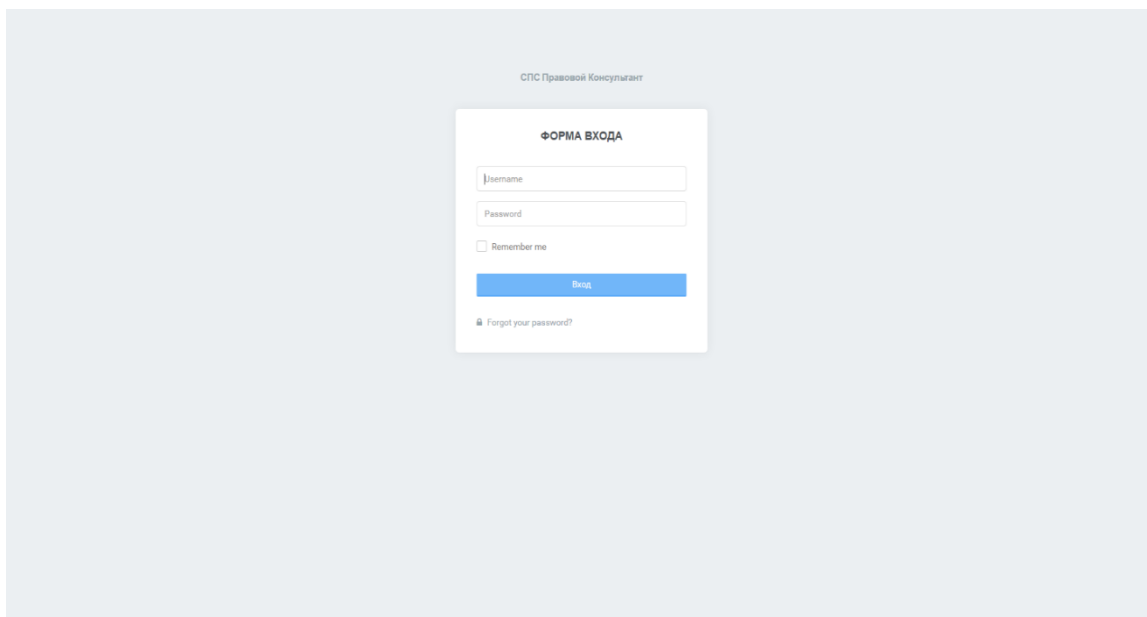


Рисунок 26 – Форма входа в панель администратора
На данном этапе происходит аутентификация и авторизация пользователей.

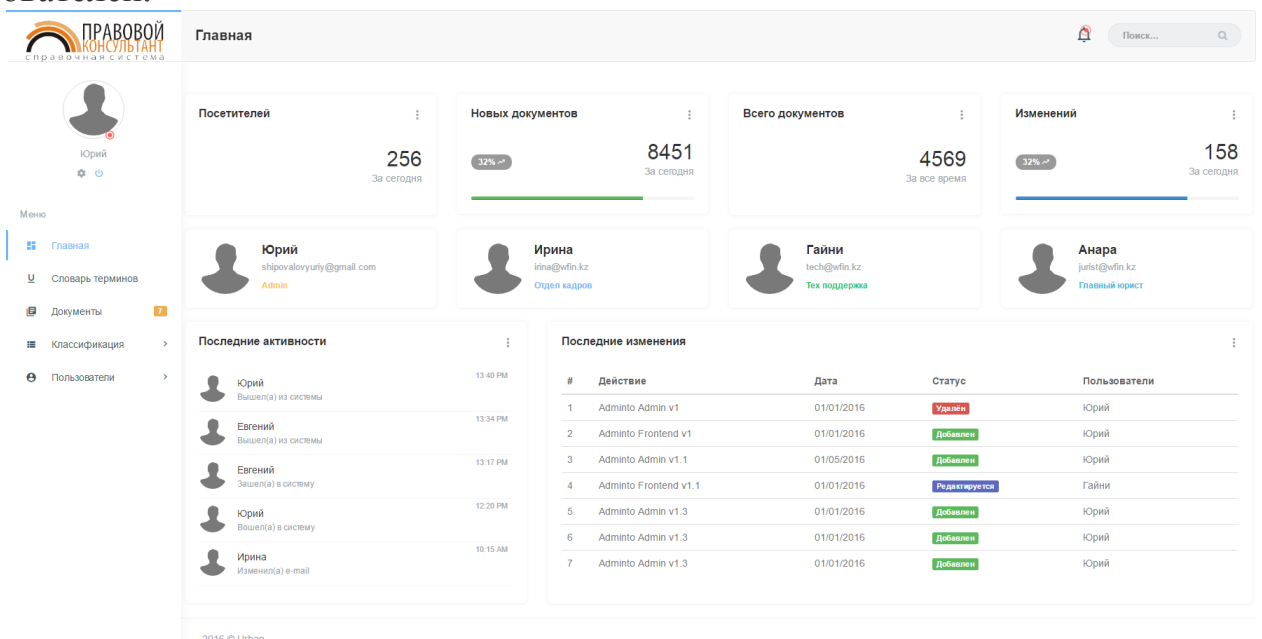


Рисунок 27 – Главная страница приложения
Главная страница приложения содержит всю необходимую информацию о последних обновлениях и активностях пользователей, меню, поиск и кнопку уведомлений.
Стоит отметить адаптивность верстки приложения

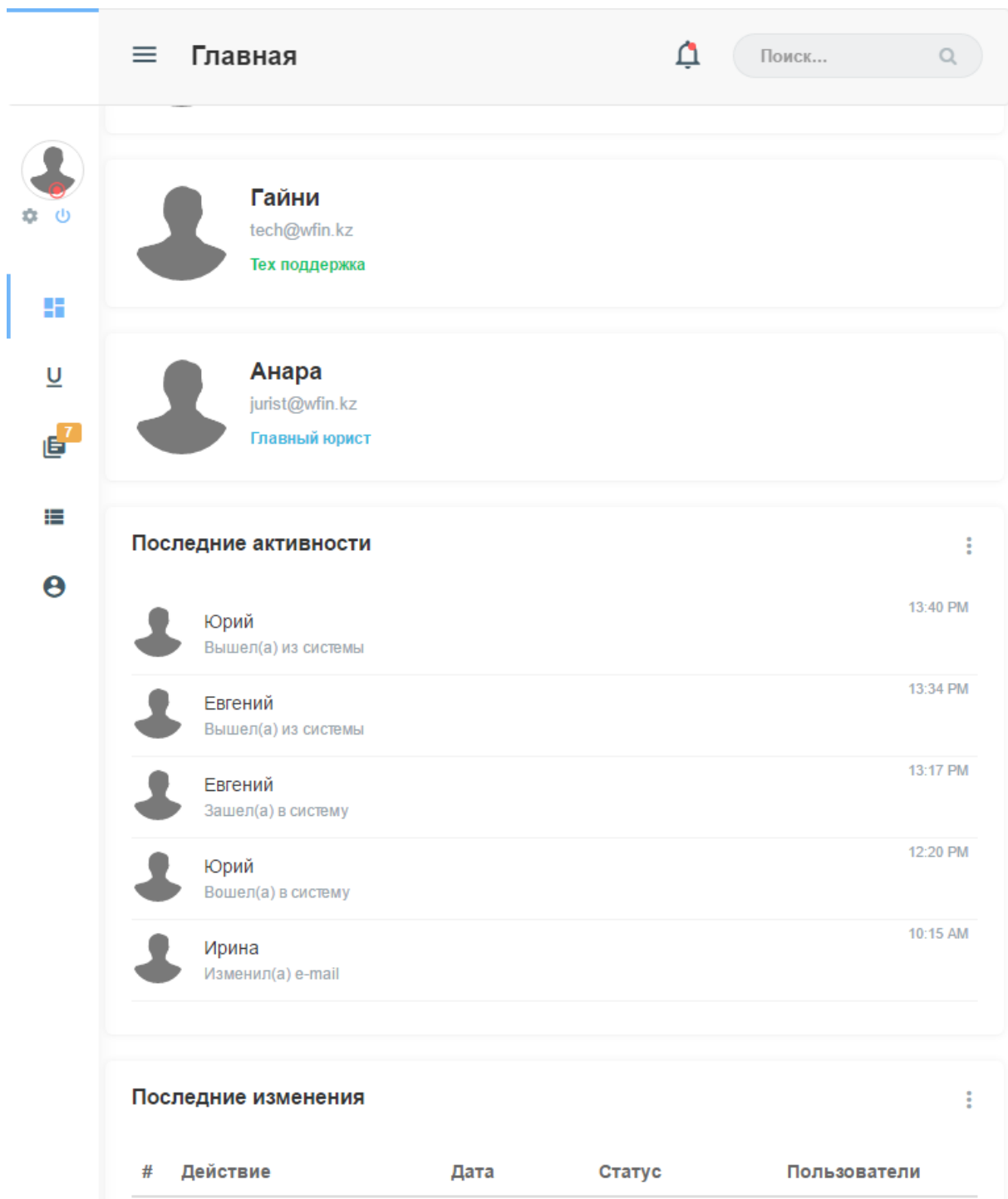


Рисунок 28 – Главная страница на экранах планшетов и мобильных устройств
Далее идет раздел «Словарь терминов»

Понятие	Дефиниция
арбитражный суд	- арбитраж court постоянно действующий третейский суд при Торгово-промышленной палате. Арбитражный суд разрешает споры, вытекающие из договорных и других гражданско-правовых отношений при осуществлении внешнеэкономических и других международных экономических связей. Решения Арбитражного суда окончательны и обжалованию не подлежат, исполняются в срок, установленный судом. Если срок исполнения не указан, они подлежат немедленному исполнению. Арбитражный суд принимает к рассмотрению споры при наличии письменного соглашения между сторонами о передаче возникшего или могущего возникнуть спора в Арбитражный суд, а также споры, подлежащие его юрисдикции. Арбитры утверждаются Президиумом ТПП на четыре года. Исчисление и распределение арбитражных споров и возмещение расходов арбитражными судами осуществляются в соответствии с положением об арбитражных сборах и расходах и об издержках сторон.
агентирование судов	- содействие администрации суда в выполнении ею своих обязанностей в порту и защита интересов судопроизводителя по его поручению и от его имени.
баллистика судебная	- отрасль криминалистической техники, изучающая огнестрельное оружие, боеприпасы к нему, следы их действия, методы и средства собирания и исследования этих объектов, а также другие технические вопросы, возникающие при расследовании преступлений, связанных с применением (ношением, хранением, изготовлением и сбытом) огнестрельного оружия и боеприпасов. Объекты исследования судебной баллистики: ручное огнестрельное оружие, отдельные части и принадлежности оружия, боеприпасы, протрады со следами применения оружия, средства и инструменты, применяемые для снаряжения патронов или изготовления снарядов, предметы со следами хранения оружия. Научную основу судебной баллистики, помимо положений криминалистики, составляют данные внешней и внутренней баллистики, оружейведения и др. областей знания.
брокерские суды	- (broker's courts) денежные суммы, занимаемые брокерами в банках или у других брокеров для различных целей. Они могут использоваться специалистами (биржевыми маклерами) для финансирования приобретения ценных бумаг, которые они продают или покупают брокерскими фирмами для финансирования гарантированного размещения новых выпусков ценных бумаг корпораций и для помощи в финансировании приобретения ценных бумаг клиентами, предпочитающими воспользоваться брокерскими кредитами при покупке ценных бумаг.
высший суд	- высший судебный орган, осуществляет надзор за судебной деятельностью всех судов общей юрисдикции. Разъяснения высшего суда обязательны для этих судов. Проверяет законность и обоснованность приговоров и решений подведомственных судов.
внешний государственный долг	- задолженность государства по непогашенным внешним займам и не выплаченным по ним процентам. Внешний государственный долг складывается из задолженности данного государства международным и государственным банкам, правительствам, частным иностранным банкам и др. Текущий внешний государственный долг - тот, по которому срок уплаты наступает в текущем или ближайшем бюджетном году; капитальный - тот, по которому срок платежа еще не наступил.
внешний долг государства	- общая сумма задолженности государства по непогашенным внешним займам и не выплаченным по ним процентам. Различают: капитальный государственный (задолженность, по которой срок платежа еще не наступил) и текущий (по которому срок уплаты наступает в данном или ближайшем бюджетном году).
выдача суд	- предоставление банком денежных средств в займы предприятиям и хозяйственным организациям для оплаты счетов за полученные ценности, оказанные услуги и выполненные работы, формирование производственных фондов и осуществление затрат
высший арбитражный суд	- центральный орган системы арбитражных судов, в ведении которых находится разбирательство дел, связанных в первую очередь с предпринимательской деятельностью. Высший арбитражный суд рассматривает наиболее сложные дела, контролирует деятельность всех арбитражных судов, дает разъяснения по правоприменению в данной области.

Рисунок 29 – Страница словаря терминов

Представлена в виде таблицы. Есть фильтрация по именованию определения, а также поиск по ключевым словам. Существует возможность добавления понятий в модальном окне.

Добавить понятие ✕

Заголовок

Дефиниция

Рисунок 30 – добавление понятия в модальном окне

Модальные окна неотъемлемая часть данного проекта, на которой я сделал акцент. Они отдельно обособлены, не отвлекают пользователя от основной части программы, экономят место на рабочем пространстве.

Раздел документов представлен удобной таблицей с набором определенных функций: добавление, удаление, редактирование, поиск.

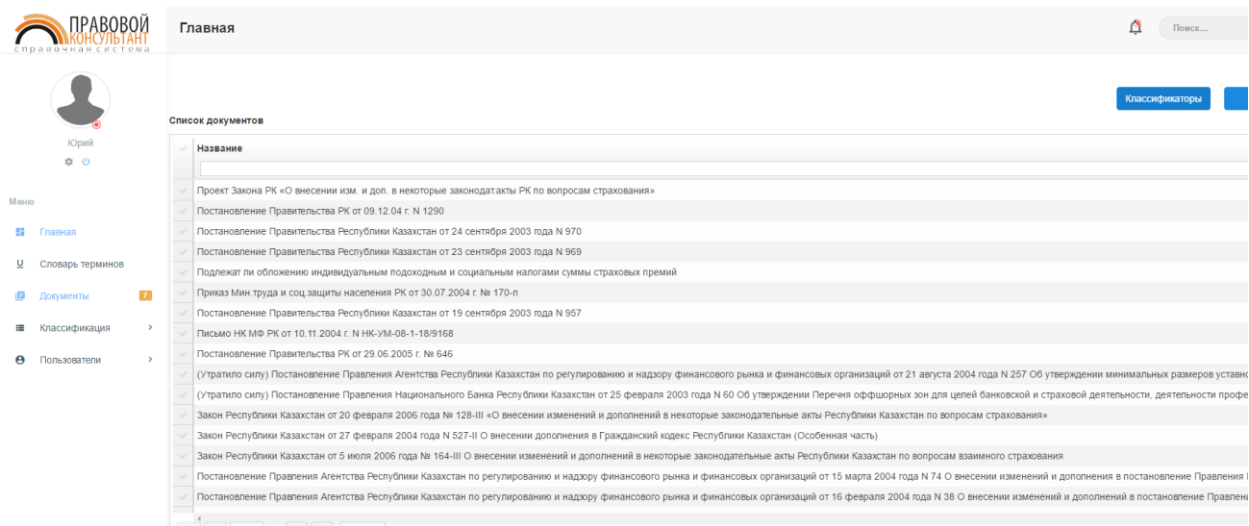


Рисунок 31 – страница раздела «Документы»
Поиск документов представлен двумя видами: поиск по классификаторам и расширенный поиск

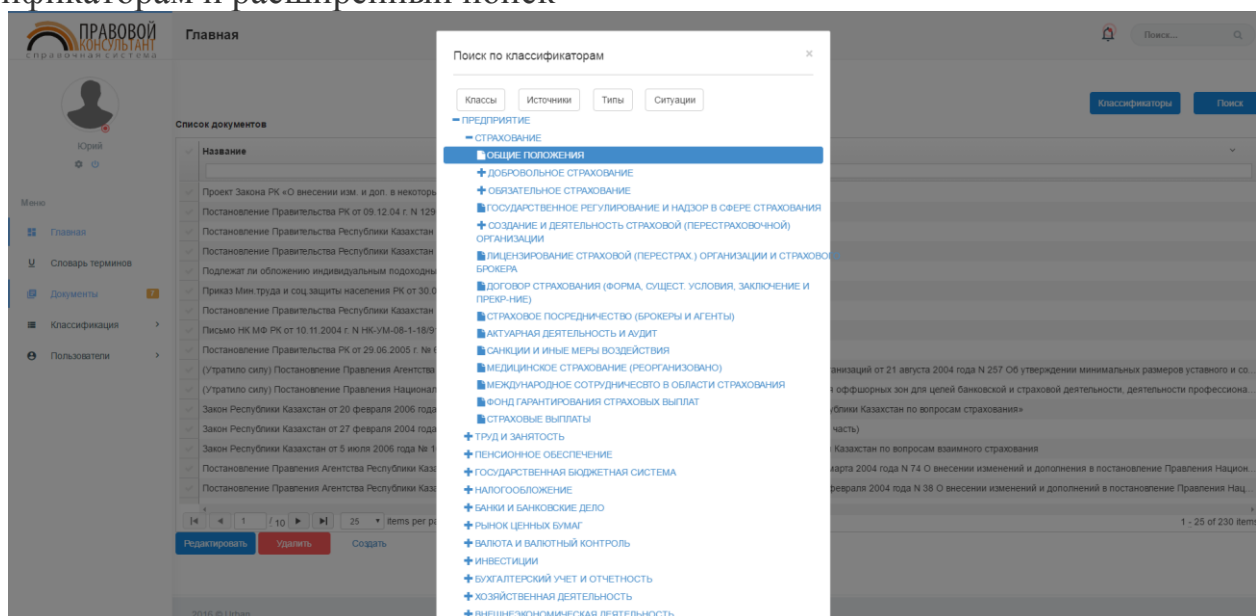


Рисунок 32 – поиск по классификаторам

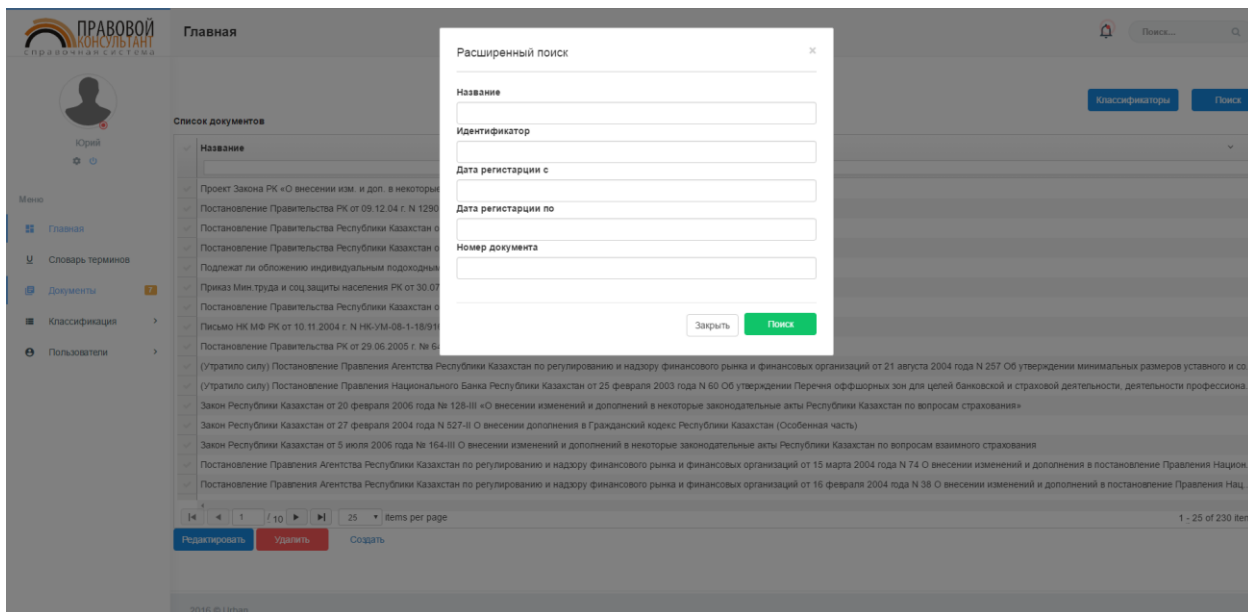


Рисунок 33 – гибкий расширенный поиск

Редактирование документов представлено удобным и функциональным редактором TinyMCE.

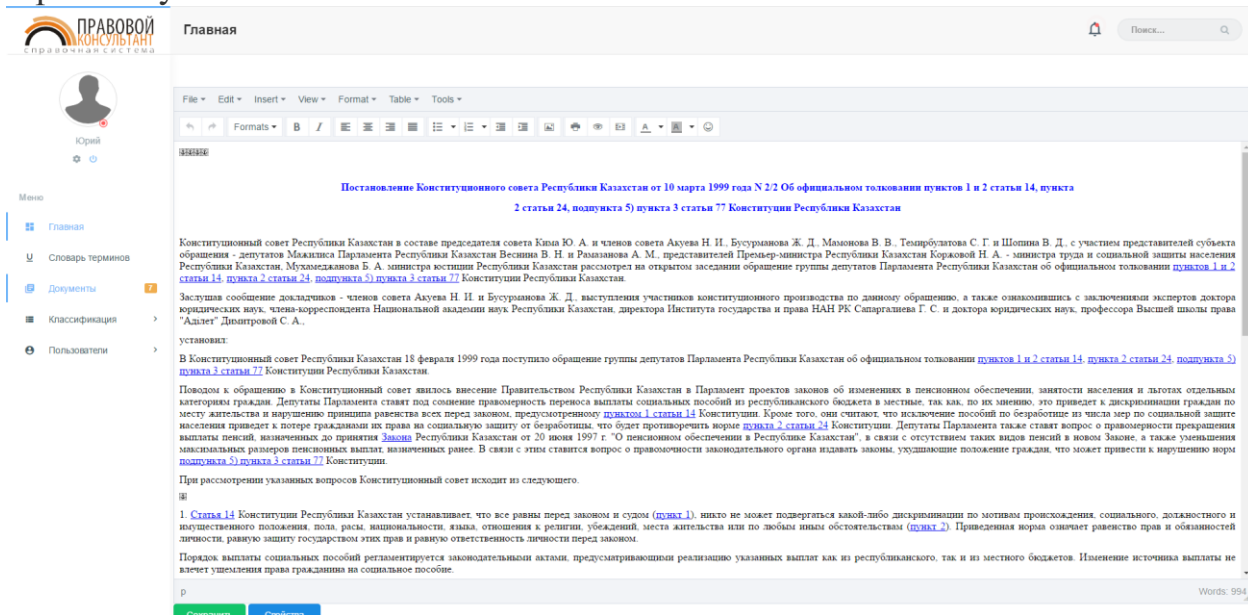


Рисунок 34 – редактор документов

Существует возможность отображать и проставлять свойства документа.

Раздел классификаторов представлен примитивным списком, позволяющим переименовывать, добавлять, удалять, сортировать пункты.

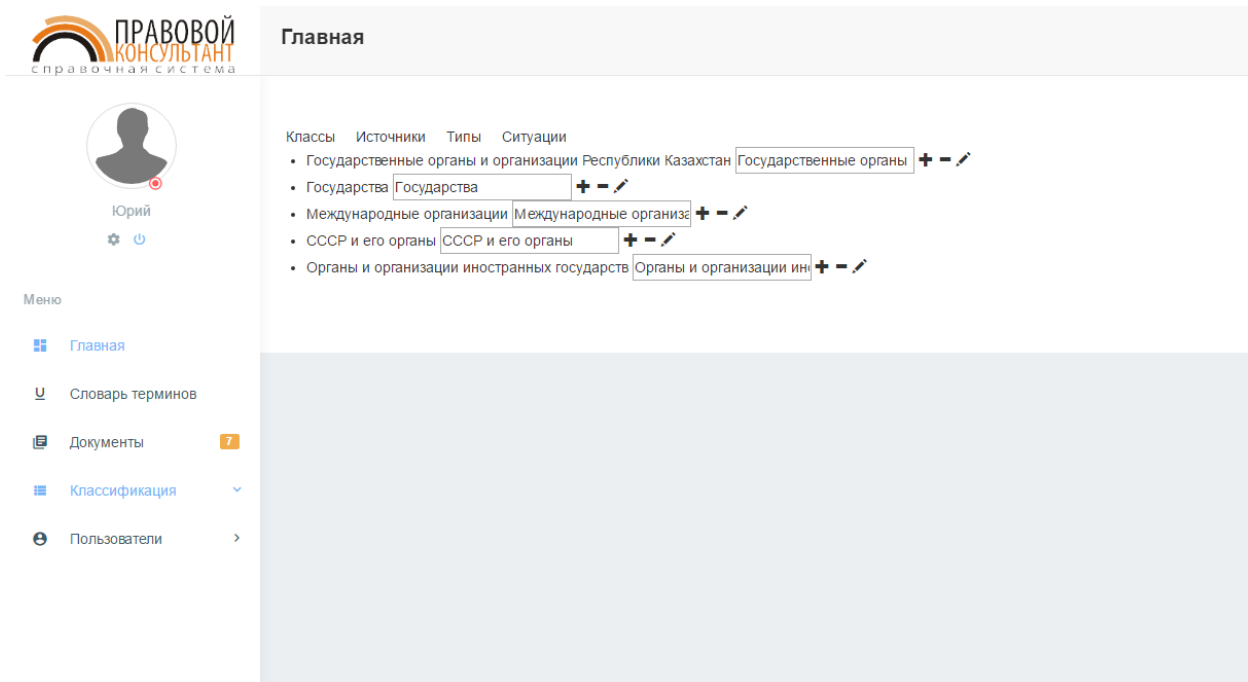


Рисунок 35 – Раздел классификаторы

Существует возможность удаления в двух режимах: удаление категории и всех ее дочерних элементов или удаление категории с указанием новой категории для дочерних элементов.

Раздел управления пользователями включает в себя список пользователей с возможностью их удаления, добавления и смены пароля.

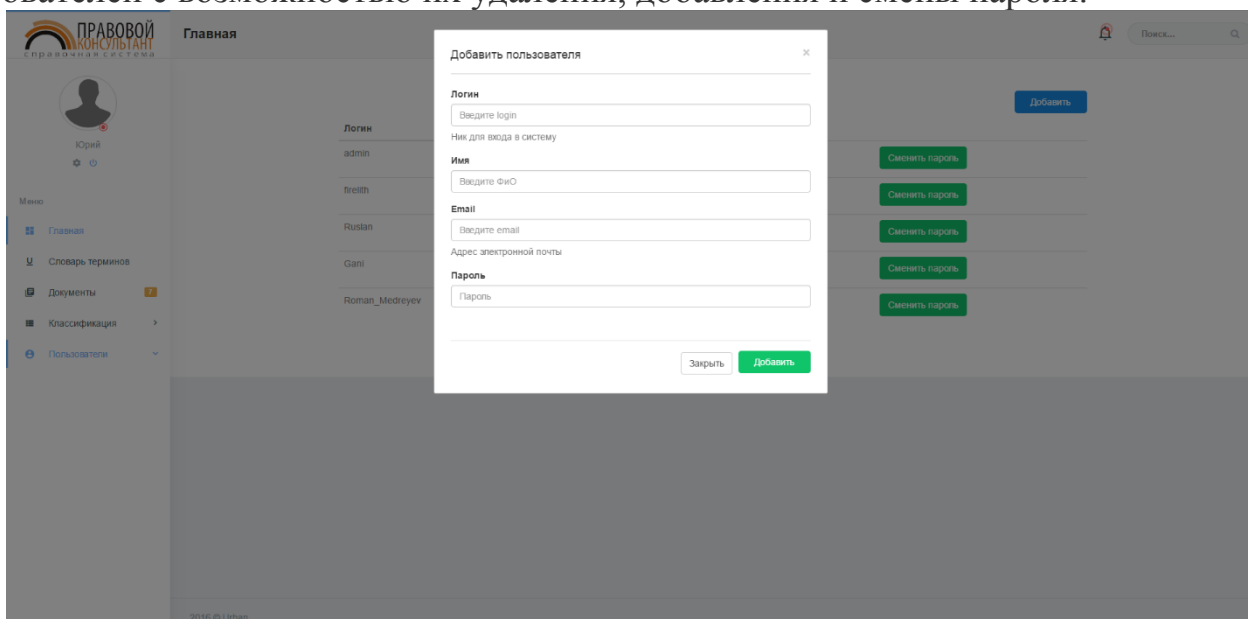


Рисунок 36 – Пример добавления пользователей в систему

Глава 5 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА

5.1 Описание работы и обоснование необходимости

Тема данной дипломной работы является «Разработка администраторской панель для системы СПС Правовой консультант»

Необходимо было разработать систему, отвечающую современным стандартам, которая уменьшила бы время обновления материалов в базе СПС Правовой консультант, сделала эффективней работу операторов ПК.

В данной главе представлена экономическая составляющая проекта, отражающая трудовые, временные и финансовый затраты в процессе реализации.

5.2 Трудовые ресурсы, используемые в работе

В проекте были задействованы менеджер, который координировал и проверял на соответствие ТЗ программу на каждом этапе ее разработки, также в проекте был задействован непосредственный исполнитель – web-разработчик

Таблица 5.1 – Данные о работниках, задействованных в проекте.

Должность	Количество человек	Заработная плата, тенге
Менеджер	1	200 000
Web-разработчик	1	250 000
Итого	2	450 000

5.3 Оборудование и программное обеспечение, используемое в работе.

Общая стоимость расходов на оборудование составляет 197 400 тенге.

Данные по количеству каждого вида оборудования и стоимости представлены в таблице 5.2.

Таблица 5.2– Используемое оборудование и программное обеспечение

Наименование материала	Единицы измерения	Количество	Цена за единицу в тенге	Сумма в тенге (без НДС)
Ноутбук HP Envy Intel(R) Core™ i7 2.2 GHz/RAM 12Gb/HDD 1000Gb	штук	1	240 000	211 200
Мышка Bloody V5	шту к	1	7 000	6 160
etBrains IntelliJ IDEA Community	шту к	1	0	0
Mysql Workbench	шту к	1	0	0
Итого:				217 360

Программное обеспечение, используемое в проекте, находится в свободном доступе и не требует лицензии.

5.4 Расчет стоимости разработки ПО

Себестоимость разработки проекта определяется по формуле, описанной ниже

$$C = \text{ФОТ} + C_{\text{н}} + A + P_{\text{а}} + C_{\text{э}} + N + C_{\text{пр}} \quad (5.1)$$

где ФОТ – фонд оплаты труда; $C_{\text{н}}$ – социальный налог; A – амортизационные отчисления; $P_{\text{а}}$ – расходы на аренду; $C_{\text{э}}$ – расходы на электроэнергию; N – накладные расходы; $C_{\text{пр}}$ – прочие расходы (использование интернета);

5.5 Сроки реализации проекта

Разработка администраторской панели осуществлялась в несколько этапов (таблица 5.3):

- 1 этап: Постановка задачи
- 2 этап: Разработка Технического задания
- 3 этап: Разработка концепции и дизайна интерфейса приложения
- 4 этап: Верстка шаблона проекта
- 5 этап: Написание логики, реализация функционала

- 6 этап: Тестирование и отладка
- 7 этап: Внедрение на предприятие
- 8 этап: Подписание акта-приема сдачи выполненных работ

Т а б л и ц а 5.3 – Этапы и сроки реализации проекта

Наименование этапа		Недели от начала работ									
											0
1 этап	Постановка задачи										
2 этап	Разработка Технического задания										
	Опрос операторов, сбор предпочтений										
	Подбор программных и языков средств разработки										
3 этап	Разработка концепции и дизайна интерфейса приложения										
4 этап	Верстка шаблона проекта										
5 этап	Написание логики, реализация функционала										
6 этап	Тестирование и отладка										
7 этап	Внедрение на предприятие										
8 этап	Приемка выполненных работ										

5.6 Расчет фонда оплаты труда

Фонд оплаты труда (ФОТ) – это суммарные издержки предприятия на оплату труда всех работников за определенный период, который определяется по формуле:

$$\text{ФОТ} = \text{З}_{\text{осн}} + \text{З}_{\text{доп}} \quad (5.2)$$

где $\text{З}_{\text{осн}}$ - основная заработная плата;

$\text{З}_{\text{доп}}$ – дополнительная заработная плата.

На этапах разработки сайта, участники разработки задействованы неравноценно, для этого необходимо рассчитать средний дневной заработок, а затем общий размер заработной платы.

Зарплата за один час рассчитывается по формуле:

$$H = \frac{\text{ЗПм}}{\text{Др} * \text{Чр}} \quad (5.3)$$

где: ЗПм – ежемесячный размер заработной платы, тенге;

Др – количество рабочих дней в месяце (21 день – пятидневная рабочая неделя);

Чр – продолжительность рабочего дня, час (при 8-часовом рабочем дне).

1) Менеджер проекта:

$$H = \frac{200000}{21 * 8} = 1190 \text{ тенге/час};$$

2) Разработчик:

$$H = \frac{250000}{21 * 8} = 1488 \text{ тенге/час};$$

Длительность цикла в днях по каждому виду работ определяется по формуле:

$$t_n = \frac{T}{q_n * z * K} \quad (5.4)$$

где T – трудоемкость этапа, норма-час;

q_n – количество исполнителей по этапу;

z – Продолжительность рабочего дня, z= 8 часов;

K – коэффициент выполнения норм времени, K = 1,1.

Полученную величину t_n округляю в большую сторону до целых дней.

$$t_1 = \frac{18}{8 * 1,1} \approx 2 \text{ дня} - \text{менеджер: постановка задачи};$$

$$t_2 = \frac{30}{8 * 1,1} \approx 4 \text{ дня} - \text{разработчик: разработка Технического задания};$$

$$t_3 = \frac{64}{8 * 1,1} \approx 8 \text{ дня} - \text{разработчик: разработка дизайна интерфейса};$$

$$t_4 = \frac{20}{8 * 1,1} \approx 3 \text{ дня} - \text{разработчик: верстка шаблона приложения};$$

$$t_5 = \frac{120}{8 * 1,1} \approx 14 \text{ дней} - \text{разработчик: разработка функционала};$$

$$t_6 = \frac{27}{8 * 1,1} \approx 3 \text{ дня} - \text{разработчик: тестирование и отладка};$$

$$t_7 = \frac{20}{8 * 1,1} \approx 3 \text{ дней} - \text{разработчик: внедрение на предприятие};$$

$$t_8 = \frac{12}{8 \times 1,1}$$

≈ 2 дней – менеджер, разработчик: приемка и сдача выполненных работ;

Сводные результаты расчета затрат на основную заработную плату работников, задействованных в разработке приложения представлены в таблице 4.6.

Таблица 5.4 – Сводные результаты расчета затрат на основную заработную плату

Наименование работ	Исполнитель	Трудоемкость		Длительность цикла, днях	Зарботная плата за час работы, тенге	Сумма заработной платы, тенге
		Нормо-час	% от общей трудоемкости			
	Менеджер	8	,8		190	2 1420
	Разработчик	0	,6		488	4 4640

	ик	Разработч	4	0,5	2	8
						488
						9
						5 232

	ик Разработч	0	,4	0	488	2 9 760
	ик Разработч	20	8,6	4	488	1 78 560

	Разработч ик	7	,7	8	3	488
						4 0 176
	Разработч ик	0	,4	6	3	488
						2 9 760

8. Приемка выполненных работ	Менедже р	2	,9	3	2	1 4 280
	Разработч ик	2	,9	3	2	1 7 856
Итого		11	00	9	3	4 71 684

Дополнительная заработная плата составляет 10 % от основной заработной платы и рассчитывается по формуле:

$$Z_{\text{доп}} = Z_{\text{осн}} * 0,1 \quad (5.5)$$

$$Z_{\text{доп}} = 471\,684 * 0,1 = 47\,168,4 \text{ тенге.}$$

Таким образом, суммарный фонд оплаты труда составит:

$$\text{ФОТ} = 471\,684 + 47\,168,4 = 518\,852,4 \text{ тенге}$$

5.7 Расчет затрат по социальному налогу

Социальный налог составляет 11% (ст. 358 п.1 НК РК) от дохода работника, и рассчитывается по формуле:

$$C_{\text{н}} = (\text{ФОТ} - \text{ПО}) * 0,11 \quad (5.6)$$

где ПО – пенсионные отчисления, которые составляют 10% от ФОТ и социальным налогом не облагаются:

$$\text{ПО} = \text{ФОТ} * 0,1 \quad (5.7)$$

$$\text{ПО} = 518\,852,4 * 0,1 = 51\,885,24 \text{ тенге.}$$

Размер отчислений на социальные нужды составит:

$$C_{\text{н}} = (518\,852,4 - 51\,885,24) * 0,11 = 51\,366,39 \text{ тенге.}$$

5.8 Расчет амортизационных отчислений

Амортизационные отчисления рассчитываются по формуле:

$$A_i = \frac{H_A * C_{пер} * N}{100 * 12 * n} \quad (5.8)$$

где H_A – норма амортизации;

$C_{пер}$ – первоначальная стоимость оборудования;

N – количество дней на выполнение работ;

n – количество дней в рабочем месяце.

Норма амортизации H_A на компьютерную технику и на программное обеспечение составляет 40% от все стоимости.

Амортизационные отчисления по используемому оборудованию и программному обеспечению составят:

$$A_1 = \frac{40 * 240\,000 * 39}{100 * 12 * 21} = 14\,857,14 \text{ тенге};$$
$$A_2 = \frac{40 * 7\,000 * 39}{100 * 12 * 21} = 433,33 \text{ тенге};$$
$$A = 14\,857,14 + 433,33 = 15\,290,47 \text{ тенге}.$$

5.9 Расчет затрат на электроэнергию

Так как, в процессе производства используется электрооборудование необходимо рассчитать затраты на электроэнергию. Затраты на электроэнергию для производственных нужд включают в себя расходы: электроэнергии на оборудование и на дополнительные нужды.

$$\mathcal{E} = \mathcal{Z}_{эл.эн.обор} + \mathcal{Z}_{доп.нуж} \quad (5.9)$$

где $\mathcal{Z}_{эл.эн.обор}$ – затраты на электроэнергию оборудования;

$\mathcal{Z}_{доп.нуж}$ – затраты электроэнергии на дополнительные нужды.

Расходы электроэнергии на оборудование рассчитываются по формуле:

$$\mathcal{Z}_{эл.эн.обор} = W * T * S * K_{исп} \quad (5.10)$$

где: W – потребляемая мощность, Вт;

T – количество часов работы оборудования;

S – стоимость киловатт-часа электроэнергии (1кВтч = 16,02 тг без НДС);

$K_{исп}$ = коэффициент использования ($K_{исп} = 0,9-1$).

$W = 90 \text{ Вт} = 0,09 \text{ кВт}$ (Мощность ноутбука);

Затраты на электроэнергию основного оборудования согласно формуле 4.11 составляет:

$$\mathcal{Z}_{эл.эн.обор} = 0,09 * 311 * 16,02 * 0,9 = 403,56 \text{ тенге}.$$

Затраты на дополнительные нужды берутся по укрупненному показателю в размере 5% от затрат на оборудование:

$$\mathcal{Z}_{доп.нуж} = 0,05 * \mathcal{Z}_{эл.эн.обор} \quad (5.11)$$

$$З_{\text{доп.нуж}} = 0,05 * 403,56 = 20,12 \text{ тенге.}$$

Суммарные затраты на электроэнергию составляют:

$$\mathcal{E} = 403,56 + 20,12 = 423,68 \text{ тенге.}$$

5.10 Расчет накладных расходов

Накладные расходы составляют 30% от ФОТ и рассчитываются по формуле:

$$НР = \text{ФОТ} * 0,3 \quad (5.12)$$

Накладные расходы согласно формуле 4.12 составляют:

$$НР = 518852,4 * 0,3 = 155\,655,72 \text{ тенге.}$$

Таким образом, в соответствии с формулой 4.1, суммарные затраты по разработке программного продукта составляют:

$$\begin{aligned} & \text{Сб} \\ & = 518852,2 + 51366,39 + 15290,47 + 100000 + 423,68 + 155655,72 + \\ & 10000 = 851588,66 \text{ тенге.} \end{aligned}$$

Смета и структура затрат по разработке комплекса лабораторных работ представлены в таблице 4.7 и на рисунке 4.1.

Т а б л и ц а 5.5 – Стоимость разработки веб-приложения.

Наименование статей затрат	Сумма, тенге	Структура затрат, %
ФОТ	518852,4	60,93
Социальный налог	51366,39	6,03
Амортизационные отчисления	15290,47	1,80
Аренда помещения	100000	11,74
Затраты на электроэнергию	423,68	0,05
Накладные расходы	155655,72	18,28
Прочие расходы	10000	1,17
Итого	851588,66	100



Рисунок 38 – Структура затрат по разработке комплекса лабораторных работ

5.11 Цена реализации

Цена реализации программного продукта складывается из его стоимости и прибыли:

$$Ц = Сб + П \quad (5.13)$$

где Сб – стоимость продукта;

П –прибыль.

При определении первоначальной цены следует задать уровень рентабельности (25%) для реализации веб-приложения:

$$Ц_n = Сб * (1+P/100) \quad (5.14)$$

где P – рентабельность.

$$Ц_n = 851588,66 * (1+0,25) = 1064485,83 \text{тенге.}$$

Цена реализации готовой продукции рассчитывается по формуле:

$$Ц_p = Ц_n + НДС \quad (5.15)$$

где НДС – налог на добавочную стоимость.

НДС рассчитывается по формуле:

$$НДС = Ц_n * 0,12 \quad (5.16)$$

$$НДС = 1064485,83 * 0,12 = 127738,3 \text{тенге;}$$

$$Ц_p = 1064485,83 + 127738,3 = 1192224,13 \text{тенге.}$$

Выводы

В результате проведенного экономического анализа проекта себестоимость с учетом накладных расходов составила 851588,66тенге. Цена реализации проекта составила 1192224,13 тенге. Большую долю расходной части занимает заработная плата рабочим 60,93% (518852,4 тенге).

Прослеживается положительный экономический эффект для организации (заказчика), так как уменьшилось время обновления и оборота документов в базе данных СПС Правовой консультант.

Нельзя не отметить личный положительный эффект, заключающийся в работе с довольно крупной системой и применением новых библиотек для разработки веб-приложения.

Глава 6. Безопасность жизнедеятельности

6.1 Анализ условий рабочего помещения

Целью данного дипломного проекта является создание веб-приложения для компании «InterConsult Info», а также предусмотреть и обеспечить необходимые требования безопасности в ходе выполнения работ.

Работа с любым цифровым устройством, в данном случае с компьютером, обусловлена высоким зрительным напряжением, вызванным излучением устройства, не стоит забывать про значительное умственное и нервно-эмоциональное напряжение. Немаловажную роль играют устройство рабочего места и интерьер, удобство которых позволит снизить нагрузку на человека и сделать его работу более комфортной.

В список опасных и вредных факторов при работе за компьютером входят:

1. повышенная температура поверхностей ПК;
2. токсические вещества;
3. повышенная или пониженная влажность воздуха;
4. отсутствие или недостаток естественного света;
5. недостаточная искусственная освещенность рабочей зоны;
6. повышенная яркость света;
7. повышенный уровень шума на рабочем месте;
8. пониженная контрастность;
9. повышенная напряженность магнитного поля;
10. нервно-эмоциональные перегрузки;
11. монотонность трудового процесса;
12. повышенный уровень статистического электричества.

Физически вредные и опасные факторы.

К телесным вредоносным и рискованным причинам относятся: завышенные значения электромагнитного, рентгеновского, ультрафиолетового и инфракрасного излучения; завышенный уровень статического электричества и запыленности воздуха рабочей зоны; завышенное содержание позитивных аэронов и пониженное содержание негативных аэроионов в атмосфере рабочей зоны; завышенный уровень блескости и ослепленности; неритмичность распределения яркости в поле зрения; завышенная насыщенность светового изображения; завышенное значение напряжения в электрической цепи, перемыкание которой имеет возможность случиться чрез тело человека.

Химически вредные и опасные факторы.

Химически вредные и небезопасные причины следующие: завышенное содержание в атмосфере рабочей зоны двуокиси углерода, озона, аммиака, фенола и формальдегида.

Психофизические вредные и опасные факторы.

Психофизиологические вредные и небезопасные причины: усилие зрения и интереса; интеллектуальные, чувственные и долгие статические перегрузки; монотонность труда; великий размер инфы, обрабатываемый в штуку медли; нерациональная организация рабочего места.

Обычными чувствами, которые чувствуют к концу рабочего дня операторы ЭВМ считаются: переутомление око, головная болезнь, тащащие боли в мышцах шейки, рук и спины, понижение сосредоточении интереса.

Магнитное поле.

Компьютер при работе создает вокруг себя электромагнитное поле, которое владеет возможностью био специфичного и теплового действия на организм человека. И за счёт воздействия электромагнитного поля на клетки и ткани человека наблюдаются нарушения символически-рефлекторной деловитости, понижение энергичности мозга. Все, наверное, имеет место быть в ведущий боли, утомляемости, смещении в худшую сторону здоровья, гипотонии. Из-за счет теплового действия электромагнитного поля увеличивается температура тела, идет нагрев тканей и органов. Более только подвержены тепловому облучению эти органы как печень, поджелудочная железа, мочевой пузырь, желудок. Имеется возможность язвы кровотечения и перфорации.

6.2 Организация и оборудование рабочего места

Рабочие места по отношению к источникам света должны располагаться так, чтобы свет падал сбоку, желательно слева.

Правильное размещение рабочих мест также немаловажно. Расстояние между соседними краями монитора должно быть 1.2 метра, в то время как расстояние между фронтальной частью монитора и тыльной частью соседнего впереди стоящего должно быть не менее 2 метров.

Дизайн и конструкция рабочего стола должны отвечать современным требованиям эргономии. Допускаются регулируемые рабочие поверхности, оптимальной высотой поверхности можно считать диапазон от 680мм до 800мм.

Глубина рабочей поверхности стола должна составлять 800 мм, но не менее 600 мм, ширина — соответственно 1 600 мм и 1 200 мм. Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной — не менее 500 мм, глубиной на уровне колен — не менее 450 мм и на уровне вытянутых ног — не менее 650 мм.

Клавиатура должна располагаться на поверхности стола на расстоянии 100-300 мм от края ближнего к пользователю.

Для точного и комфортного восприятия информации монитор должен находится ниже горизонтали уровня взгляда на 30 градусов, при этом расстояние от него до глаз должно быть не менее 50 см. При работе за ПК

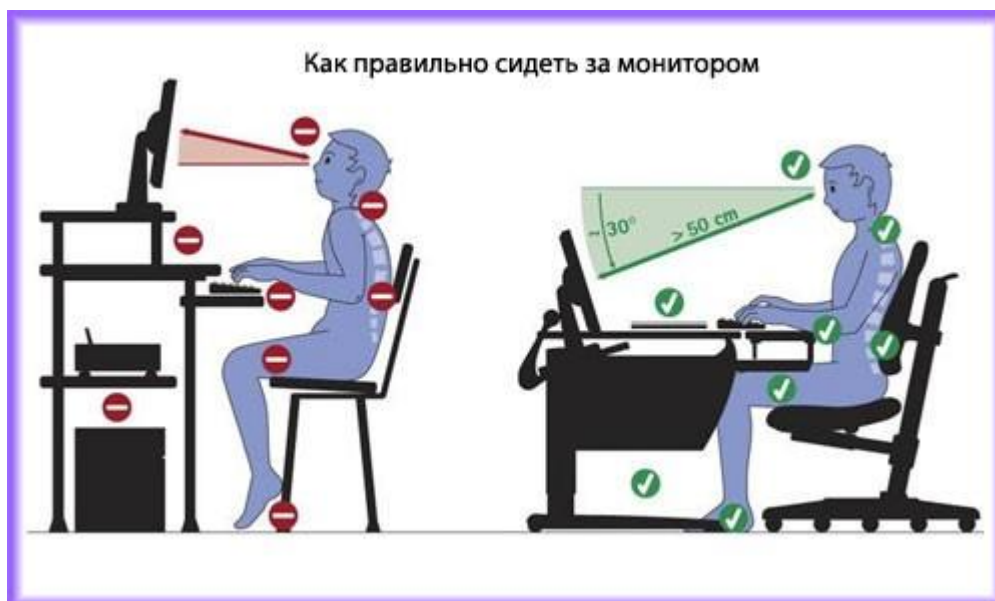
важную роль играет положение тела человека и возможность периодически его смены во избежание затеканий и других негативных последствий. Это достигается использованием настраиваемых положений высоты и наклона кресел.

Конструкция стула должна обеспечивать:

- ширину и глубину поверхности сиденья не менее 400 мм;
- отсутствие острых углов у поверхности сиденья
- регулировку высоты поверхности сиденья в пределах 400-550 мм и углом наклона вперед до 15 градусов и назад до 5 градусов.;
- высоту опорной поверхности спинки 300 ± 20 мм, ширину — не менее 380 мм и радиус кривизны горизонтальной плоскости 400 мм;
- угол наклона спинки в вертикальной плоскости в пределах 0 ± 30 градусов;
- регулировку расстояния спинки от переднего края сидения в пределах 260-400 мм;
- стационарные или съемные подлокотники длиной не менее 250 мм и шириной 50-70 мм;
- регулировку подлокотников по высоте над сиденьем в пределах 230 ± 30 мм и внутреннего расстояния между подлокотниками в пределах 350-500 мм.;
- поверхность сиденья, спинки и подлокотников должна быть полумягкой, с нескользящим неэлектризующимся, воздухопроницаемым покрытием, легко очищаемым от загрязнения.

Рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 град.

На рисунке 1 показано оптимальное и нежелательное расположение тела в момент работы на ПК, а также правильную структуру рабочего места.



6.3 Организация труда и отдыха при работе за компьютером

Режим труда и отдыха при работе за компьютером регламентируется в зависимости от вида и продолжительности трудовой деятельности на ПК.

Виды трудовой деятельности на ПК разделяются на 3 группы: группа А - работа, связанная со считыванием информации; группа Б — работа по вводу информации; группа В — работа в режиме постоянного взаимодействия с ПК.

Основным и приоритетным видом деятельности будет тот, которым рабочий занимается более 50% своего рабочего времени.

Таблица 6.1 - Виды категорий трудовой деятельности с ПК

Категория работы по тяжести и напряженности	Уровень нагрузки за рабочую		
	Группа А Количество знаков	Группа Б Количество знаков	Группа В Время работы, ч
I	До 20000	До 15000	До 2,0
II	До 40000	До 30000	До 4,0
III	До 60000	До 40000	До 6,0

При 8-часовой рабочей смене и работе на ПК регламентированные перерывы следует устанавливать:

- для первой категории работ через 2 часа от начала смены и через 2 часа после обеденного перерыва продолжительностью 15 минут каждый;
- для второй категории работ — через 2 часа от начала рабочей смены и через 1,5-2,0 часа после обеденного перерыва продолжительностью 15 минут каждый или продолжительностью 10 минут через каждый час работы;
- для третьей категории работ — через 1,5- 2,0 часа от начала рабочей смены и через 1,5-2,0 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

При 12-часовой рабочей смене регламентированные перерывы должны устанавливаться в первые 8 часов работы аналогично перерывам при 8-часовой рабочей смене, а в течение последних 4 часов работы, независимо от категории и вида работ, каждый час продолжительностью 15 минут.

Продолжительность непрерывной работы на ПК без регламентированного перерыва не должна превышать 2 часа.

6.4 Освещение

1) Следует избегать большого контраста между яркостью экрана и окружающего пространства. Оптимальным считается их выравнивание.

2) Запрещается работа с компьютером в темном или полутемном помещении, Освещение в помещениях с ПК должно быть смешанным: естественным – за счет солнечного света – и искусственным.

В качестве источников общего искусственного освещения лучше всего использовать осветительные приборы, которые создают равномерную освещенность путем рассеянного или отраженного светораспределения (свет от ламп падает непосредственно на потолок) и исключают блики на экране монитора и клавиатуре. В соответствии с санитарными нормами, это должны быть преимущественно люминесцентные лампы типа ЛБ с рассеивателями или экранирующими решетками. Пульсации света люминесцентных ламп действуют раздражающе на зрение и нервную систему операторов, поэтому для уменьшения коэффициента пульсации используйте лампы, укомплектованные высокочастотными пускорегулирующими аппаратами. Следует отметить, что существуют специальные люминесцентные лампы, например, фирмы «VitaLight R», которые излучают свет различного «качества», имитируя, таким образом, полный спектр естественного солнечного света. Эти лампы меньше раздражают, чем любые другие лампы искусственного освещения.

Источники света необходимо равномерно распределять по комнате, komponуя в сплошные или прерывистые линии. Линии должны располагаться сбоку от рабочих мест параллельно линии зрения пользователя – при рядомном размещении компьютеров; локализовано над рабочим столом – при размещении рабочих мест по периметру помещения. Грамотная организация освещения способна повысить производительность труда при зрительной работе средней трудности – на 5-6%, при очень трудной – на 15%.

Электростатическое поле, вредные вещества в воздухе

При работе компьютер образует вокруг себя электростатическое поле, которое деионизирует окружающую среду, а при нагревании платы и корпус монитора испускают в воздух вредные вещества. Всё это делает воздух очень сухим, слабо ионизированным, со специфическим запахом и в общем "тяжёлым" для дыхания. Естественно, такой воздух не может быть полезен для организма и может привести к заболеваниям аллергического характера, болезням органов дыхания и другим расстройствам.

6.5 Расчет освещенности помещения

Расчет освещенности рабочего места играет немаловажную роль, при плохом освещении снижается производительность труда, ухудшается зрение персонала.

Для рационального освещения необходимо выполнение следующих условий:

- постоянная освещенность рабочих поверхностей;
- достаточная и равномерно распределенная яркость освещаемых рабочих поверхностей;
- отсутствие резких контрастов между яркостью рабочей поверхностью и окружающего пространства;
- отсутствие в поле зрения светящихся поверхностей, обладающих большим блеском, что достигается применением светильников с рассеянным светом и увеличением высоты их подвеса.

Для рационального освещения производственных помещений и рабочих мест большое значение имеет выбор цвета для окраски потолка, стен и производственного оборудования.

Рассчитаем люминесцентное освещение кабинета, предназначенного для выполнения работ с размером объекта различения от 0,3 мин до 1 мин. Размеры помещений: $A=5$ м, $B=4$ м, $H=3$ м. Площадь помещения определяем по формуле:

$$S=A*B \quad (6.5.1)$$

Подставив значения получим:

$$S=4*3=20 \text{ м}^2$$

Освещение проектируется при помощи светильников ОД минимальной освещенностью $E_{\text{мин}}=200$ лк, средней удельной мощностью 17-23 Вт/м². Высота подвеса над рабочей поверхностью $H_p=2$ м. Освещение выполняется лампами ЛБ-2*40, световой поток ламп $F_{\text{л}}=2480$ лм, длина 1.2 м, коэффициент запаса равен $\kappa=1.5$.

Определим показатель помещения по формуле:

$$F=A*B/H_p(A+B) \quad (6.5.2)$$

Подставив значения получим:

$$F=5*4/2*(5+4)=1.1$$

Затем, для $F=1.1$, коэффициентов отражения потолка $P_n=0,7$, стен $P_c=0,5$ и расчетной плоскости $P_p=0,3$ находим коэффициент использования светового потока $h=0,51$. Потребное число светильников определяется по формуле:

$$N=E_{\text{мин}}*S*\kappa/F_{\text{л}}*h*Z*n \quad (6.5.3)$$

где $n=2$ общее число ламп в светильнике.

Подставив значения получим:

$$N=200*20*1,5/2480*0,51*1.1*2=2 \text{ штуки.}$$

Общее количество ламп равно $n=2*2=4$.

Таким образом, выполнение перечисленных выше требований к организации рабочего места обеспечит комфортные условия умственного труда.

6.6 Расчет уровня шума

Высокий уровень шума, создаваемый ПК периферией неблагоприятен для персонала. Он может быть причиной возникновения головных болей, потерей внимания и коммуникации между рабочими. Поэтому необходимо привести уровень шума на рабочем месте в соответствие с регламентом.

Суммарный уровень шума и величина его давления рассчитывается как сумма величин давления отдельных некогерентных источников шума:

$$L_{\Sigma} = 10 \lg \sum_{i=1}^{i=n} 10^{0,1L_i}$$

(6.6.1)

где L_i - уровень звукового давления i -го источника шума;

n - количество источников шума.

Полученный результат необходимо сопоставить с допустимым значением уровня шума для данного помещения и в случае превышения допустимой величины принять ряд мер для уменьшения текущего шума. Это достигается за счет установки звукопоглощающих корпусов на компьютер и периферию, очистки от пыли и грязи функциональных частей устройств, за счет рационального расположения устройств на рабочем пространстве.

Уровни звукового давления источников шума, действующих на оператора на его рабочем месте представлены в табл. 2

Таблица 6.2 – Средний уровень некоторых источников шума

Источник шума	Уровень шума, дБ
Жесткий диск	40
Вентилятор	45
Монитор	17
Клавиатура	10
Принтер	45
Сканер	42

Как правило, источниками шума от ПК являются его составные такие как жесткий диск, вентиляторы на процессоре и видеокарте, монитор и другие. Подставив средние уровни звукового давления каждого устройства получим:

$$L_{\Sigma} = 10 \cdot \lg(10^4 + 10^{4,5} + 10^{1,7} + 10^1 + 10^{4,5} + 10^{4,2}) = 49,5 \text{ дБ}$$

Полученное значение не превышает допустимый уровень шума для рабочего места оператора, равный 65 дБ (ГОСТ 12.1.003-83). Если учесть тот факт, что данное значение было получено при условии, что все устройства будут работать одновременно, то можно сделать вывод что фактический шум на рабочем месте будет ниже полученной величины.

Выводы.

Безопасность любой деятельности человека и окружающей среды должна рассматривать с учетом всех экономических, социальных и экологических последствий. В последнее время в связи с демографическим кризисом и развитием технологий уменьшилось количество рабочих на предприятиях. В целях повышения производительности труда внедряются новые революционные технологии, производится модернизация уже существующих процессов и все же не стоит забывать об обеспечении безопасности рабочих. В данной главе мы ознакомились с необходимыми требованиями безопасности при работе на ПК. Ознакомились и рассчитали немаловажные показатели как уровень шума и освещенность, убедились, что они соответствуют принятым стандартам.

Заключение

Данная дипломная работа посвящена разработке панели администратора для СПС «Правовой консультант». На основе проблем предыдущей панели и требований, выдвинутых заказчиком, были сформированы следующие цели:

1. Разработать техническое задание;
2. Разработать клиентскую часть и интерфейс;
3. Разработать серверную часть приложения;
4. Отладить и внедрить.

По ходу выполнения дипломного проекта получил опыт в работе с популярным фреймворком AngularJS, актуальный в настоящее время в промышленном программировании. Реализовал первый транш задач, поставленных заказчиком. Система работает и функционирует без ошибок и багов. Проект выполнен в виде веб-приложения с учетом современных тенденций дизайна и промышленного программирования. Также в ходе выполнения работы был проведен технико-экономический анализ приложения и безопасности жизнедеятельности.

В результате внедрения административной панели в производство уменьшилось время обновления системы, вырос спрос продукта на рынке, увеличилась эффективность работы операторов.

Список литературы

1. Фурсов В.Г. Финансовый менеджмент. Москва, 2010.
2. Голубицкая Е.А. Экономика связи: М.: - Ирмас, 2006.
3. Базылов К.Б., Алибаева С.А., Бабич А.А. Выпускная работа бакалавров. Экономический раздел. Методические указания для студентов всех форм обучения специальности 050719 –Радиотехника электроника и телекоммуникации. – Алматы: АИЭС, - 2008.
4. Хакимжанов Т.Е. Сборник задач по охране труда и безопасности жизнедеятельности: Учебное пособие для вузов. – Алматы: Эверо, 2007. – 274СНиП РК 2.04-05-2002. Естественное и искусственное освещение.
5. Общие требования. – Алматы, 2002. Сивков В.П., Смирнов С.Г., Козьяков А.Ф. и др. Сборник типовых
6. расчетов по курсу «Охрана труда». – М.: МВТУ, 1979. – 79с
7. Безопасность жизнедеятельности / Под ред. Н.А. Белова. – М.: Знание,
8. Адам Фримен “ASP .NET MVC 4 с примерами на С# 5.0”
9. Документация Microsoft msdn/microsoft.com
- 10.Мишель Дамлер Microsoft SQL Server 2008
- 11.<https://ru.wikipedia.org/wiki/HTML5>
- 12.[https://ru.wikipedia.org/wiki/Bootstrap_\(%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA\)](https://ru.wikipedia.org/wiki/Bootstrap_(%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA))
- 13.[Кит Джереми - HTML5 для веб-дизайнеров](#)
- 14.[П. Лабберс - HTML 5 для профессионалов](#)
- 15.[Кристофер Шмитт, Кайл Симпсон - HTML5. Рецепты программирования](#)
- 16.Дэвид Макфарланд "Большая книга CSS3" (2014, PDF)

Приложение А

```
package kz.novelty.jurist;
import SevenZip.*;
import SevenZip.LZMAHelper;
import com.google.common.base.Joiner;
import net.sf.json.JSON;
import net.sf.json.JSONArray;
import net.sf.json.JSONObject;
import org.apache.noggit.JSONParser;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.*;
import java.sql.*;
import java.text.MessageFormat;
import java.util.*;
import java.util.logging.Logger;

import static kz.novelty.jurist.Connector.Instance;

/**
 * Created by aun on 02.02.2016.
 */
@WebServlet(name = "adminRequest",urlPatterns="/adminPanel")

public class adminRequest extends HttpServlet {
    private Connection con;
    private Statement st;
    private ResultSet res;
    StringBuilder struct = new StringBuilder();
    interface handler {
        void process(HttpServletRequest req, HttpServletResponse resp, final
        OutputStream out) throws ServletException, IOException, SQLException;
    }
}
```

```

HashMap<String, handler> handlers = new HashMap<String, handler>();

public adminRequest (){
    super();
    handlers.put("getCategories", new handler() {
        public void process(HttpServletRequest req, HttpServletResponse resp,
OutputStream out) throws ServletException, IOException, SQLException {
            process_getCategory(req,resp,out);
        }
    });
    handlers.put("getClasses", new handler() {
        public void process(HttpServletRequest req, HttpServletResponse resp,
OutputStream out) throws ServletException, IOException, SQLException {
            process_getClasses(req,resp,out);
        }
    });
    handlers.put("getTerms", new handler() {
        public void process(HttpServletRequest req, HttpServletResponse resp,
OutputStream out) throws ServletException, IOException, SQLException {
            process_getTerms(req,resp,out);
        }
    });
    handlers.put("addTerm", new handler() {
        public void process(HttpServletRequest req, HttpServletResponse resp,
OutputStream out) throws ServletException, IOException, SQLException {
            process_addTerm(req,resp,out);
        }
    });
    handlers.put("delTerms", new handler() {
        public void process(HttpServletRequest req, HttpServletResponse resp,
OutputStream out) throws ServletException, IOException, SQLException {
            process_delTerm(req,resp,out);
        }
    });
    handlers.put("getAllCategories", new handler() {
        public void process(HttpServletRequest req, HttpServletResponse resp,
OutputStream out) throws ServletException, IOException, SQLException {
            process_getAllCat(req,resp,out);
        }
    });
    handlers.put("getDocs", new handler() {
        public void process(HttpServletRequest req, HttpServletResponse resp,
OutputStream out) throws ServletException, IOException, SQLException {
            process_getDocs(req,resp,out);
        }
    });
}

```

```

    }
    });
    handlers.put("getDoc", new handler() {
        public void process(HttpServletRequest req, HttpServletResponse resp,
        OutputStream out) throws ServletException, IOException, SQLException {
            process_docLoad(req,resp,out);
        }
    });
    handlers.put("getDocInfo", new handler() {
        public void process(HttpServletRequest req, HttpServletResponse resp,
        OutputStream out) throws ServletException, IOException, SQLException {
            process_docInfo(req,resp,out);
        }
    });
    handlers.put("getDocTree", new handler() {
        public void process(HttpServletRequest req, HttpServletResponse resp,
        OutputStream out) throws ServletException, IOException, SQLException {
            process_docTree(req,resp,out);
        }
    });
    handlers.put("searchDocs", new handler() {
        public void process(HttpServletRequest req, HttpServletResponse resp,
        OutputStream out) throws ServletException, IOException, SQLException {
            process_searchDocs(req,resp,out);
        }
    });
    handlers.put("getUserList", new handler() {
        public void process(HttpServletRequest req, HttpServletResponse resp,
        OutputStream out) throws ServletException, IOException, SQLException {
            process_getUserList(req,resp,out);
        }
    });
}

```

```

public void process_getUserList( HttpServletRequest req, HttpServletResponse
resp,OutputStream out) throws IOException, SQLException {

```

```

    JSONArray data = (JSONArray)
Connector.Instance().executeQuery("SELECT * FROM jurist.users c join
jurist.user_roles b on b.user_id = c.user_id where role_id = 1", new Object[]{} , new
DBConnector.QueryToJsonObjectHandler(), Boolean.TRUE, struct);
    resp.setContentType("application/json; charset=UTF-8");
    out.write(data.toString().getBytes("UTF-8"));

```

```

}

public void process_getTerms( HttpServletRequest req, HttpServletResponse
resp,OutputStream out) throws SQLException, IOException {

    String title = req.getParameter("title");
    JSONArray data = (JSONArray)
Connector.Instance().executeQuery("SELECT * FROM baza.terms where word like
'%" +title+"%'";, new Object[] {}, new DBConnector.QueryToJsonObjectHandler(),
Boolean.TRUE, struct);
    resp.setContentType("application/json; charset=UTF-8");
    out.write(data.toString().getBytes("UTF-8"));

}

public void process_addTerm(HttpServletRequest req, HttpServletResponse resp,
OutputStream out){

}

public void process_delTerm(HttpServletRequest req, HttpServletResponse
resp,OutputStream out){

}

public void process_searchDocs(HttpServletRequest req, HttpServletResponse
resp,OutputStream out) throws SQLException, IOException {

    String id = req.getParameter("id");
    String shortName = req.getParameter("shortName");
    String regDate = req.getParameter("regDate");
    String regDateUntil = req.getParameter("regDateUntil");
    String code = req.getParameter("code");

    List<String> strings = new ArrayList<String>();

    if ((id != null)&&(id != "")) { strings.add("b.id like "+id);}
    if ((shortName != null)&&(shortName != "")) { strings.add("b.fullName like
'%" +shortName+"%");}
    if ((code != null)&&(code != "")) { strings.add("c.code like '"+code+"");}
    if ((regDate != null)&&(regDateUntil != null)) { strings.add("b.publishDate
between '"+regDate+"";strings.add("'+regDateUntil+'");}
    if ((regDateUntil == null)&&(regDate != null)) { strings.add("b.publishDate >=
'+regDate+'");}
    if ((regDateUntil != null)&&(regDate == null)) { strings.add("b.publishDate <=
'+regDateUntil+'");}

```

```

String criteria = Joiner.on(" and ").join(strings);

String query = "select * from baza.properties b join baza.codes c on c.docId =
b.id where "+criteria;

JSONArray data = (JSONArray) Connector.Instance().executeQuery(query,
new Object[]{} , new DBConnector.QueryToJsonObjectHandler(), Boolean.TRUE,
struct);
resp.setContentType("application/json; charset=UTF-8");
out.write(data.toString().getBytes("UTF-8"));

}
public void process_getCategory(HttpServletRequest req, HttpServletResponse
resp , OutputStream out) throws SQLException, IOException {

    ResultSet res = null;
    String class_id = req.getParameter("id");
    JSONArray data = (JSONArray) Connector.Instance().executeQuery("select *
from baza.trees where class_id = "+class_id+"", new Object[]{} , new
DBConnector.QueryToJsonObjectHandler(), Boolean.TRUE, struct);
    resp.setContentType("application/json; charset=UTF-8");
    out.write(data.toString().getBytes("UTF-8"));

}
public void process_getAllCat(HttpServletRequest req, HttpServletResponse resp
, OutputStream out) throws SQLException, IOException {

    ResultSet res = null;
    JSONArray data = (JSONArray) Connector.Instance().executeQuery("select *
from baza.trees", new Object[]{} , new DBConnector.QueryToJsonObjectHandler(),
Boolean.TRUE, struct);
    resp.setContentType("application/json; charset=UTF-8");
    out.write(data.toString().getBytes("UTF-8"));

}

public void process_getClasses(HttpServletRequest req,HttpServletResponse
resp, OutputStream out) throws SQLException, IOException {
    ResultSet res = null;
    JSONArray data = (JSONArray) Connector.Instance().executeQuery("select *
from baza.classificators", new Object[]{} , new
DBConnector.QueryToJsonObjectHandler(), Boolean.TRUE, struct);

```



```

        resp.setContentType("application/json; charset=UTF-8");
        out.write(data.toString().getBytes("UTF-8"));
    }

    public void process_getDocs(HttpServletRequest req, HttpServletResponse resp,
        OutputStream out) throws SQLException, IOException {

        ResultSet res = null;
        String tree_id = req.getParameter("id");
        JSONArray data = (JSONArray) Connector.Instance().executeQuery("select *
        from baza.properties b join baza.tree_document c on c.doc_id = b.id where c.tree_id
        =" + tree_id + "'", new Object[] { }, new DBConnector.QueryToJsonObjectHandler(),
        Boolean.TRUE, struct);
        resp.setContentType("application/json; charset=UTF-8");
        out.write(data.toString().getBytes("UTF-8"));

    }

    protected void process_docLoad(HttpServletRequest req, HttpServletResponse
    resp, final OutputStream out) throws ServletException, IOException,
    SQLException {
        String id = req.getParameter("id");
        String data = (String) Connector.Instance().executeQuery("select body from
        baza.documents where id=" + id, new Object[] { }, new Connector.QueryHandler() {
            public Object onQuery(ResultSet rs, Object... params) throws
            SQLException {
                String val = "";
                if (rs.next()) {
                    try {
                        ByteArrayOutputStream bs = new ByteArrayOutputStream();
                        bs.reset();
                        LZMAHelper.Decompress(rs.getBinaryStream("body"), bs);
                        val = bs.toString("utf-8");
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
                return val;
            }
        });
        out.write(data.getBytes("UTF-8"));
    }
}

```

```

protected void process_docInfo(HttpServletRequest req, HttpServletResponse
resp, final OutputStream out) throws ServletException, IOException,
SQLException {
    String id = req.getParameter("id");
    try {
        JSONArray data = (JSONArray) Connector.Instance().executeQuery("select
p.* from baza.properties p where p.id=?", new Object[]{id}, new
DBConnector.QueryToJsonObjectHandler(), Boolean.TRUE, struct);
        resp.setContentType("application/json; charset=UTF-8");
        out.write(data.toString().getBytes("UTF-8"));
    } catch (SQLException e) {
        e.printStackTrace();
        throw new ServletException(e);
    }
}

protected void process_docTree(HttpServletRequest req, HttpServletResponse
resp, final OutputStream out) throws ServletException, IOException, SQLException
{

```

```

    String id = req.getParameter("id");
    ResultSet res = null;
    JSONArray data = (JSONArray)
Connector.Instance().executeQuery("SELECT * FROM baza.trees a\n" +
    "join baza.tree_document b on a.id = b.tree_id\n" +
    "where b.doc_id ="+id+";", new Object[]{} , new
DBConnector.QueryToJsonObjectHandler(), Boolean.TRUE, struct);
    resp.setContentType("application/json; charset=UTF-8");
    out.write(data.toString().getBytes("UTF-8"));

}

```

```

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {

```

```

    String rtype = req.getParameter("requestType");
    handler handler = handlers.get(rtype);
    OutputStream out = resp.getOutputStream();
    if (handler==null){
        resp.setStatus(500);
        //resp.sendError(500, " UNKNOWN REQUEST " + rtype);
        out.write((" UNKNOWN REQUEST " + rtype).getBytes());
        out.flush();
        out.close();
    }
}

```

```

    }else
    try {
        handler.process(req,resp,out);
    } catch (SQLException e) {
        e.printStackTrace();
    }

    out.flush();
    out.close();

}

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {

    doPost(req,resp);

}

public void print(OutputStream stream, String s) throws IOException {
    if (s == null) s = "null";
    int len = s.length();
    for (int i = 0; i < len; i++) {
        char c = s.charAt(i);
        if ((c & 65280) != 0) {
            ResourceBundle lStrings =
ResourceBundle.getBundle("javax.servlet.LocalStrings");
            String errMsg = lStrings.getString("err.not_iso8859_1");
            Object errArgs[] = new Object[1];
            errArgs[0] = new Character(c);
            errMsg = MessageFormat.format(errMsg, errArgs);
            throw new CharConversionException(errMsg);
        }
        stream.write(c);
    }
}

}

<!DOCTYPE html>
<html>

```

<!-- Mirrored from coderthemes.com/admintto_1.2/light/ by HTTrack Website Copier/3.x [XR&CO'2014], Mon, 23 May 2016 09:33:24 GMT -->

`<head>`

`<meta charset="utf-8">`

`<meta name="viewport" content="width=device-width, initial-scale=1.0">`

`<link rel="shortcut icon" href="assets/images/favicon.ico">`

`<title>Правовой консультант - администратор</title>`

<!-- Morris Chart CSS -->

`<link rel="stylesheet" href="assets/plugins/morris/morris.css">`

<!-- App css -->

`<link href="assets/css/bootstrap.min.css" rel="stylesheet" type="text/css" />`

`<link href="assets/css/core.css" rel="stylesheet" type="text/css" />`

`<link href="assets/css/components.css" rel="stylesheet" type="text/css" />`

`<link href="assets/css/icons.css" rel="stylesheet" type="text/css" />`

`<link href="assets/css/pages.css" rel="stylesheet" type="text/css" />`

`<link href="assets/css/menu.css" rel="stylesheet" type="text/css" />`

`<link href="assets/css/responsive.css" rel="stylesheet" type="text/css" />`

`<link rel="stylesheet" href="assets/js/angular-ui-grid/ui-grid.css">`

`<link rel="stylesheet" href="assets/style.css">`

`<link rel="stylesheet"`

`href="//netdna.bootstrapcdn.com/bootstrap/3.0.3/css/bootstrap.min.css">`

`<link rel="stylesheet" href="assets/js/angular-bootstrap-nav-tree/dist/abn_tree.css"/>`

<!-- HTML5 Shiv and Respond.js IE8 support of HTML5 elements and media queries -->

<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

<!--[if lt IE 9]>

<script

src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>

<script

src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>

<![endif]-->

`<script src="assets/js/modernizr.min.js"></script>`

`</head>`

```

<body class="fixed-left">

<!-- Begin page -->
<div ng-app="mainApp" id="wrapper">

  <!-- Top Bar Start -->
  <div class="topbar">

    <!-- LOGO -->
    <div class="topbar-left">
      <a href="#" class="logo"><i class="zmdi zmdi-layers"></i></a>
    </div>

    <!-- Button mobile view to collapse sidebar menu -->
    <div class="navbar navbar-default" role="navigation">
      <div class="container" style="width:100%;">

        <!-- Page title -->
        <ul class="nav navbar-nav navbar-left">
          <li>
            <button class="button-menu-mobile open-left">
              <i class="zmdi zmdi-menu"></i>
            </button>
          </li>
          <li>
            <h4 class="page-title">Главная</h4>
          </li>
        </ul>

        <!-- Right(Notification and Searchbox -->
        <ul class="nav navbar-nav navbar-right">
          <li>
            <!-- Notification -->
            <div class="notification-box">
              <ul class="list-inline m-b-0">
                <li>
                  <a href="javascript:void(0);" class="right-bar-toggle">
                    <i class="zmdi zmdi-notifications-none"></i>
                  </a>
                  <div class="noti-dot">
                    <span class="dot"></span>
                    <span class="pulse"></span>
                  </div>
                </li>
              </ul>
            </div>
          </li>
        </ul>
      </div>
    </div>
  </div>

```

```
        </li>
      </ul>
    </div>
    <!-- End Notification bar -->
  </li>
  <li class="hidden-xs">
    <form role="search" class="app-search">
      <input type="text" placeholder="Поиск..."
        class="form-control">
      <a href="#"><i class="fa fa-search"></i></a>
    </form>
  </li>
</ul>

</div><!-- end container -->
</div><!-- end navbar -->
</div>
<!-- Top Bar End -->

<!-- ===== Left Sidebar Start ===== -->
<div class="left side-menu">
  <div class="sidebar-inner slimscrollleft">

    <!-- User -->
    <div class="user-box">
      <div class="user-img">
        
        <div class="user-status offline"><i class="zmdi zmdi-dot-
circle"></i></div>
      </div>
      <h5><a href="#">Юрий</a> </h5>
      <ul class="list-inline">
        <li>
          <a href="#" >
            <i class="zmdi zmdi-settings"></i>
          </a>
        </li>

        <li>
          <a href="#" class="text-custom">
            <i class="zmdi zmdi-power"></i>
          </a>
        </li>
      </ul>
    </div>
  </div>
</div>
```

```
</li>
</ul>
</div>
<!-- End User -->

<!-- Sidemenu -->
<div id="sidebar-menu">
  <ul>
    <li class="text-muted menu-title">Меню</li>

    <li>
      <a href="#" class="waves-effect active"><i class="zmdi zmdi-view-
dashboard"></i> <span> Главная </span> </a>
    </li>

    <li>
      <a href="#terms" class="waves-effect"><i class="zmdi zmdi-format-
underlined"></i> <span>Словарь терминов</span> </a>
    </li>

    <li class="has_sub">
      <a href="#doc" class="waves-effect"><i class="zmdi zmdi-
collection-text"></i><span class="label label-warning pull-right">7</span><span>
Документы </span> </a>
    </li>

    <li class="has_sub">
      <a href="#category" class="waves-effect"><i class="zmdi zmdi-
view-list"></i> <span> Классификация </span> <span class="menu-
arrow"></span></a>
    </li>

    <li class="has_sub">
      <a href="#users" class="waves-effect"><i class="zmdi zmdi-
account-circle"></i><span> Пользователи </span> <span class="menu-
arrow"></span></a>
    </li>

  </ul>
  <div class="clearfix"></div>
</div>
<!-- Sidebar -->
```

```
<div class="clearfix"></div>
```

```
</div>
```

```
</div>
```

```
<!-- Left Sidebar End -->
```

```
<!--
```

```
=====  
===== -->
```

```
<!-- Start right Content here -->
```

```
<!--
```

```
=====  
===== -->
```

```
<div class="content-page">
```

```
<!-- Start content -->
```

```
<div class="content">
```

```
<div class="container" style="width:100%;">
```

```
<div style="margin-top:50px;" class="row">
```

```
<div ng-view></div>
```

```
</div>
```

```
</div> <!-- container -->
```

```
</div> <!-- content -->
```

```
<footer class="footer text-right">
```

```
2016 © Urban.
```

```
</footer>
```

```
</div>
```

```
<!--
```

```
=====  
===== -->
```

```
<!-- End Right content here -->
```

```
<!--
```

```
=====  
===== -->
```



```
<!-- Right Sidebar -->
<div class="side-bar right-bar">
  <a href="javascript:void(0);" class="right-bar-toggle">
    <i class="zmdi zmdi-close-circle-o"></i>
  </a>
  <h4 class="">УВЕДОМЛЕНИЯ</h4>
  <div class="notification-list nicescroll">
    <ul class="list-group list-no-border user-list">
      <li class="list-group-item">
        <a href="#" class="user-list-item">
          <div class="avatar">
            
          </div>
          <div class="user-desc">
            <span class="name">Michael Zenaty</span>
            <span class="desc">There are new settings available</span>
            <span class="time">2 hours ago</span>
          </div>
        </a>
      </li>
      <li class="list-group-item">
        <a href="#" class="user-list-item">
          <div class="icon bg-info">
            <i class="zmdi zmdi-account"></i>
          </div>
          <div class="user-desc">
            <span class="name">New Signup</span>
            <span class="desc">There are new settings available</span>
            <span class="time">5 hours ago</span>
          </div>
        </a>
      </li>
      <li class="list-group-item">
        <a href="#" class="user-list-item">
          <div class="icon bg-pink">
            <i class="zmdi zmdi-comment"></i>
          </div>
          <div class="user-desc">
            <span class="name">New Message received</span>
            <span class="desc">There are new settings available</span>
            <span class="time">1 day ago</span>
          </div>
        </a>
      </li>
    </ul>
  </div>
</div>
```

```
</a>
</li>
<li class="list-group-item active">
  <a href="#" class="user-list-item">
    <div class="avatar">
      
    </div>
    <div class="user-desc">
      <span class="name">James Anderson</span>
      <span class="desc">There are new settings available</span>
      <span class="time">2 days ago</span>
    </div>
  </a>
</li>
<li class="list-group-item active">
  <a href="#" class="user-list-item">
    <div class="icon bg-warning">
      <i class="zmdi zmdi-settings"></i>
    </div>
    <div class="user-desc">
      <span class="name">Settings</span>
      <span class="desc">There are new settings available</span>
      <span class="time">1 day ago</span>
    </div>
  </a>
</li>
</ul>
</div>
</div>
<!-- /Right-bar -->

</div>
<!-- END wrapper -->

<script>
  var resizefunc = [];
</script>

<!-- jQuery -->
<script src="assets/js/jquery.min.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
```

```
<script src="assets/js/detect.js"></script>
<script src="assets/js/fastclick.js"></script>
<script src="assets/js/jquery.blockUI.js"></script>
<script src="assets/js/waves.js"></script>
<script src="assets/js/jquery.nicescroll.js"></script>
<script src="assets/js/jquery.scrollTo.min.js"></script>
<!------->
```

```
<script src="assets/js/angular/angular.js" charset="utf-8"></script>
<script src="assets/js/angular-cookies/angular-cookies.js" charset="utf-8"></script>
<script src="assets/js/angular-route/angular-route.js" charset="utf-8"></script>
<script src="assets/js/angular-resource/angular-resource.js" charset="utf-
8"></script>
<script src="assets/js/angular-ui-grid/ui-grid.min.js" charset="utf-8"></script>
<script src="assets/js/app/main.js" charset="utf-8"></script>
<script src="assets/js/app/controllers/loginController.js"></script>
<script src="assets/js/app/services/userLogin.js"></script>
<script src="assets/js/app/controllers/userController.js" charset="utf-8"></script>
<script src="assets/js/app/controllers/termsController.js" charset="utf-8"></script>
<script src="assets/js/app/controllers/documentController.js" charset="utf-
8"></script>
<script src="assets/js/app/controllers/documentViewController.js" charset="utf-
8"></script>
<script src="assets/js/app/services/termsFactory.js" charset="utf-8"></script>
<script src="assets/js/app/directives/termDer.js" charset="utf-8"></script>
<script src="assets/js/app/directives/catDir.js" charset="utf-8"></script>
<script src="assets/js/app/controllers/editorController.js" charset="utf-8"></script>
<script src="assets/js/app/services/editorFac.js" charset="utf-8"></script>
<script src="assets/js/app/services/catFactory.js" charset="utf-8"></script>
<script src="assets/js/app/services/docView.js" charset="utf-8"></script>
<script src="assets/js/app/services/Document.js" charset="utf-8"></script>
<script src="assets/js/node_modules/lzma/src/lzma.js" charset="utf-8"></script>
<script src="assets/js/app/filters/classFilter.js"></script>
<script src="assets/js/app/controllers/category.js"></script>
<script src="assets/js/angular-drag-and-drop-lists/angular-drag-and-drop-
lists.js"></script>
<script src="http://angular-ui.github.io/bootstrap/ui-bootstrap-tpls-
0.9.0.js"></script>
<script src="assets/js/angular-bootstrap-nav-
tree/dist/abn_tree_directive.js"></script>
```

```
<!-- KNOB JS -->
<!--[if IE]>
```

```
<script type="text/javascript" src="assets/plugins/jquery-
knob/excanvas.js"></script>
<![endif]-->
<script src="assets/plugins/jquery-knob/jquery.knob.js"></script>
```

```
<!--Morris Chart-->
<script src="assets/plugins/morris/morris.min.js"></script>
<script src="assets/plugins/raphael/raphael-min.js"></script>
<script src="assets/pages/jquery.morris.init.js"></script>
```

```
<!-- Dashboard init -->
<script src="assets/pages/jquery.dashboard.js"></script>
```

```
<!-- App js -->
<script src="assets/js/jquery.core.js"></script>
<script src="assets/js/jquery.app.js"></script>
```

```
</body>
```

```
<!-- Mirrored from coderthemes.com/admintto_1.2/light/ by HTTrack Website
Copier/3.x [XR&CO'2014], Mon, 23 May 2016 09:34:05 GMT -->
</html>
```

```
'use strict';
```

```
mainApp.controller('categoryController', function ($scope, $rootScope,
$modal,$route, Category) {
```

```
    Category.getClass().then(function (data) {
```

```
        $scope.classificators = data;
```

```
    });
```

```
    $scope.getTree = function(id){
```

```
    Category.getTree(id).then(function (data) {
```

```
        console.log(data);
```

```
        $scope.classItems = data;
```

```
    });
```

```

}
$scope.addCat = function (classifier, category, el) {
  Category.add(classifier, category, el);
  var obj = {'id': '', 'name': el, 'class': classifier.id, 'children': []}
  category.children.push(obj);
};
$scope.delCat = function (index, el, current) {
  $scope.modalInstance2 = null;
  var modalInstance = $modal.open({
    templateUrl: 'view/category/dialog.html',
    controller: 'modalDialogController',
    size: 'sm',
    resolve: {
      deleteType: function () {
        return $scope.deleteType;
      },
      classItems: function(){
        return $scope.classItems;
      }
    }
  });
  var deleteOne = function () {
    current.splice(index, 1);
  }

  modalInstance.result.then(function (deleteType) {
    switch (deleteType.data) {
      case 'this':
        deleteOne();
        break;
      case 'select':
        deleteType.selected.children =
Array.prototype.concat(el.children,deleteType.selected.children);
        current.splice(index,1);
        $route.reload();
        break;
    }
  });
};
$scope.replace = function (index, el, list) {
  var next_list = jQuery(target_list).scope().classItem;
  list.splice(index, 1);
  Category.replace(el, next_list);
};

```

```

    }

    $scope.editCat = function (el, item) {

        Category.edit(el, item);
        item.name = el;

    };

});

mainApp.controller('modalDialogController', function ($scope,
$modalInstance, classItems) {
    $scope.deleteType = {data: "",
selected:"
    };
    $scope.classItems = classItems;

    $scope.ok = function () {
        $modalInstance.close($scope.deleteType);
    };

    $scope.cancel = function () {

        $modalInstance.dismiss('cancel');

    };
});

'user strict';
mainApp.directive('edited', function () {

return {

    link: function (scope, element, attr) {

        element.val(scope.classItem.name);

        element.on('keypress', function ($event) {
            if ($event.which == 13) {
                scope.editCat(element.val(), scope.classItem);
            }
        });
    }
};
});

```

```

    });

    }

  });
});

mainApp.directive('added', function ($compile) {

  return {

    link: function ($scope, element, attr) {

      element.on('click', function () {
        var el = element.parent().find('ul:first');
        var y = angular.element($compile("<li><input ng-model='adding'
type='text'/></li><span class='glyphicon glyphicon-ok' ng-
click='addCat(classificator,classItem,adding)'></span>")($scope));
        var x = el.prepend(y);
      });
      //element.on('keypress',function($event){
      //
      // if( $event.which==13){
      //   console.log('asdfsadf');
      // }
      //})
    }
  }
});
var current;
mainApp.directive('replaced',function($compile){

  return {
    scope:true,
    link:function(scope,element,attr){

      element.on('dragend',function(){

        $compile(element)(scope);

      });
    }
  }
}

```



```

        tree:tree,
        children: getChildren(collection, category.id,tree)
    }
    })
}
});
},
add: function (classifier, category, el) {
    console.log(category.id);
    var opt = {
        url: 'assets/js/app/data/category.php',
        method: "GET",
        params: {'class_id': classifier.id, 'parent_id': category.id, 'name': el}
    }
    return $http(opt).then(function (response) {

        console.log(response.data);

    });
},
delete: function (el,children) {

},
edit: function (el, item) {
    var opt = {
        url: 'assets/js/app/data/category.php',
        method: "GET",
        params: {'el_id': item.id, 'new_title': el}
    }
    return $http(opt)

},
replace:function(el,next_parent){
    //console.log(el,last_parent,next_parent);
    var opt = {
        url: 'assets/js/app/data/category.php',
        method: "GET",
        params: {'el': el.id,'next_parent':next_parent.id}
    }
}

```