

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

по направлению 02.03.03 Математическое обеспечение и
администрирование информационных систем

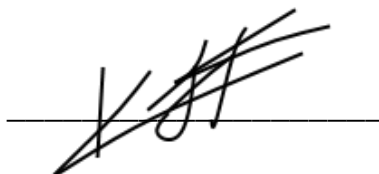
тип программы академическая

профиль «Администрирование информационных систем»

Разработка системы учета и автоматизированного заполнения карточек
образцов подписей государственных информационных систем

Студент

Группа АИ-16



Кургузов К.И.

Руководитель

канд. техн. наук, доцент



Ведищев В.В.

Консультант по

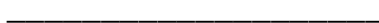
программному обеспечению

канд. техн. наук, доцент



Ведищев В. В.

Нормоконтроль



Болдырихин О.В.

Заведующий кафедрой

канд. техн. наук



Алексеев В.А.

Липецк 2020 г.

ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет ФАИ

Кафедра АСУ

Заведующий кафедрой Алексеев В.А.

« » _____ 20 ____ г.

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Студенту Кургузову Кириллу Игоревичу группы АИ-16

Направление (специальность) 09.03.01 «Математическое обеспечение и администрирование информационных систем»

1. Тема Разработка системы учета и автоматизированного заполнения карточек образцов подписей государственных информационных систем.

2. Цель и задачи работы Повышение эффективности деятельности учреждения, за счёт ускорения поиска карточек образцов подписей и пропадание необходимости участвовать некоторых людей в бизнес-процессах, в которых они незаинтересованно, путем внедрения в государственное учреждение Управление Федерального Казначейства по Липецкой области систему хранения карточек образцов подписей с историями их изменений и автоматизированным заполнением.

3. Характеристика предметной области упрощение хранения и доступа к информации, хранящейся в государственных учреждениях

4. Содержание расчетно-пояснительной записки введение, постановка задачи, изучение и моделирование предметной области, разработка информационной базы, программно-аппаратная реализация, представление и

анализ полученных результатов, заключение (выводы), список источников

5. Перечень графического материала иллюстрации предметной области, схема функциональной структуры, диаграмма "сущность-связь", физическая структура базы данных, структура аппаратного обеспечения, структура программного обеспечения

6. Срок сдачи ВКР руководителю 18.06.2020

7. Консультанты по ВКР _____

8. Дата выдачи задания 10.02.2020

9. Руководитель ВКР

канд. техн. наук, доцент Ведищев В.В.

10. Задание принял к исполнению студент Кургузов К. И.

Заведующий кафедрой АСУ Алексеев В.А.

Аннотация

С. 49. Ил. 20. Литература 3 назв. Прил. 2.

В рамках данной выпускной работы проведено теоретическое и экспериментальное исследование проблемы хранения информации в государственных учреждениях.

На основе выясненной информации разработано соответствующее программное обеспечение.

Оглавление

Введение.....	7
1 Постановка задачи	9
1.1 Литературный и патентный обзор постановки подобных задач.....	9
1.2 Объекты управления, информационные объекты и автоматизируемые процессы. Пользователи и внешние сущности.....	10
1.3 Цели разработки, функции системы, ограничения и критерии оценки результатов.....	11
1.3.1 Цели разработки	11
1.3.2 Функции системы.....	11
1.3.3 Ограничения системы.....	11
2. Изучение и моделирование предметной области	12
2.1 Выявление основных понятий и процессов, их свойств и закономерностей. Построение ER-диаграммы предметной области.....	12
2.2 Теоретическое изучение предметной области. Построение теоретических математических моделей	14
2.3 Экспериментальное изучение предметной области. Построение эмпирических математических моделей.....	15
3.1 Построение концептуальной и физической модели данных.....	17
3.2 Описание источников информации, входных сигналов и документов	22
3.3 Описание выходной информации: сигналов, документов и видеок кадров.....	22
4. Программно-аппаратная реализация решения задачи	23
4.1 Аппаратное обеспечение	23
4.2 Программное обеспечение	23

4.3 Разработанные программные средства	26
4.3.2 Описание программы.....	31
4.3.2.1 Общие сведения	31
4.3.2.2 Функциональное назначение	32
4.3.2.3 Описание логической структуры	32
4.3.2.4 Используемые технические средства.....	34
4.3.2.7 Выходные данные	35
4.3.3 Описание применения программы	36
4.3.3.1 Назначение программы	36
4.3.3.2 Условия применения.....	36
4.3.3.3 Описание задачи.....	36
4.3.3.4 Входные и выходные данные	36
4.3.4 Описание результатов работы программы.....	37
5 Результаты внедрения и использования системы. Достижение целей разработки	38
Заключение	39
Список литературы	40
Приложение А. Шаблон карточки образцов подписей.....	41
Приложение Б. Интерфейс приложения.	43

Введение

В информационном обществе электронный документ становится важнейшей формой информационного обмена между гражданами, организациями и органами власти. Для улучшения эффективности государственного управления, качества и оперативности предоставляемых государственных услуг необходимо совершенствовать механизмы информационного обмена.

Документы закрепляют производственные отношения, как внутри предприятия, так и с другими организациями и нередко служат письменным доказательством при возникновении имущественных, трудовых и иных споров.

В процессе работы УФК требуется большое количество карточек образцов подписей. После их использования в оперативной деятельности для решения текущих вопросов карточки становятся хранителями ретроспективной информации, надобность в которой может возникнуть вновь через определенное время.

Использование карточек по прошествии времени затруднительно и времязатратно, т.к. быстро найти нужную не всегда возможно, ситуация осложняется тем, что УФК хранит историю изменений карточек и после нужно проверить актуальность информации.

Дела с добавлением в систему новой карточки обстоят ничуть не лучше. В этом процессе задействуются люди в обязанности которых не входит выполнение данной функции. Операторы отправляют письмо по служебной почте, в котором вручную набирают содержимое карточки образцов подписи и отсканированную физическую версию, системным администраторам, чтобы те загрузили полученные данные на сервер.

Актуальность выбранной темы обуславливается недостаточной проработкой проблемы хранения документов в государственных учреждениях.

Объектом темы является деятельность государственного учреждения

Управление Федерального Казначейства по Липецкой области.

Предметом является карточки образцов подписи, образующиеся в результате работы учреждения.

1 Постановка задачи

1.1 Литературный и патентный обзор постановки подобных задач

В целях решения задачи оказания государственных услуг и ускорения обработки запросов граждан, в настоящее время создаются информационные ресурсы по документам органов исполнительной власти различных периодов [1].

На данный момент существуют различные варианты программных продуктов для хранения документов. Например, такие системы, как:

E1 ЕФВРАТ «Система электронного архива»

Компании КАДФЕМ Си-Ай-Эс внедрила систему электронного документооборота "Е1 Евфрат" компании Cognitive Technologies [2]. Внедрение данной системы позволило заметно сократить рутинные процессы документооборота, что в свою очередь, привело к ускорению темпов работы всех отделов компании. Также, отмечаются улучшение показателей эффективности управленческих процессов и улучшение качества ведения делопроизводства в целом. В Е1 реализован весь необходимый функционал для организации электронного архива. Также, Е1 обеспечивает ведение единой базы данных, исключающей дублирование документов и информации, что позволяет значительно экономить время как на регистрации и копировании, так и на их транспортировке между удаленными структурными подразделениями. Система имеет собственную встроенную систему сканирования и распознавания, в основе которой лежит система оптического распознавания текстов Cuneiform. Сохранение и сжатие распознанного документа происходит с помощью технологии Cognitive PDF/A, что позволяет уменьшить его размер в 10 раз без потери качества. Поскольку компания КАДФЕМ Си-Ай-Эс имеет большое количество филиалов, то благодаря использованию системы "Е1 Евфрат", можно наблюдать повышение уровня оперативности взаимодействия с удаленными офисами.

ЭЛАР «Саперион»

Единая платформа для создания систем сбора, обработки, хранения и управления документами организации. Саперион обеспечивает хранение неограниченного объема контента любого вида: скан-образов бумажных документов, файлов офисных приложений, медиа и файлов специализированных приложений. Архитектура ядра поддерживает территориально распределенную конфигурацию хранилища и включает в себя сервисы репликации данных и балансировки аппаратной нагрузки. Поиск документов в системах, содержащих более миллиарда документов, осуществляется за секунды. Пользователи могут использовать сквозной поиск по всем разделам (с учетом прав доступа), индексным полям, контекстный и полнотекстовый поиск, справочники и рубрикаторы. Подсистема ввода позволяет напрямую подключить сканирующие устройства, в том числе скоростные потоковые сканеры, индексировать документы, распознавать штрихкоды, баркоды и полные тексты документов. Кроме того, платформа обладает функциями пакетной обработки изображений: автоматический доворот по тексту, обрезка полей, исключение пустых листов и листов разделителей.

1.2 Объекты управления, информационные объекты и автоматизируемые процессы. Пользователи и внешние сущности

Пользователи системы:

1. Сотрудники отдела ведения федеральных реестров, в обязанности которых входят добавление, изменение и просмотр карточек.
2. Сотрудники отдела режима секретности и безопасности информации, в обязанности которых входят только просмотр карточек.
3. Сотрудники других подразделений, в обязанности которых входят только просмотр карточек.
4. Администратор информационной системы, в обязанности которого входит регистрация новых пользователей.

1.3 Цели разработки, функции системы, ограничения и критерии оценки результатов.

1.3.1 Цели разработки

Целью разработки данной работы является повышение эффективности деятельности учреждения, за счёт ускорения поиска карточек образцов подписей и пропадание необходимости участвовать некоторым людям в бизнес-процессах, в которых они незаинтересованно, путем внедрения в государственное учреждение Управление Федерального Казначейства по Липецкой области систему хранения карточек образцов подписей с историями их изменений и автоматизированным заполнением.

1.3.2 Функции системы

Для реализации целей в пункте 1.3.1 необходимо чтобы система поддерживала следующий функционал:

1. Регистрация или аутентификация пользователя.
2. Просмотр информации карточек образцов подписей.
3. Поиск карточки по заданным параметрам.
4. Добавление новой карточки.
5. Просмотр истории изменения карточек образцов подписей.
6. Ведение журнала событий.

1.3.3 Ограничения системы

Система ограничена возникновением проблем с доступом к интернету.

2. Изучение и моделирование предметной области

2.1 Выявление основных понятий и процессов, их свойств и закономерностей. Построение ER-диаграммы предметной области

В ходе прохождения практики в УФК, были выявлены основные процессы. Первоначально администратор регистрирует пользователей, имеющих доступ к карточкам образцов подписей, указывая их должность. Операторы авторизуются с помощью логинов и паролей, выданных администратором. Далее оператор может просмотреть, найти, добавить, удалить нужную ему карточку. Иногда сотрудникам отдела безопасности нужно получить данные хранящиеся в карточке.

Обобщенная use-case диаграмма представлена на рисунке 1.

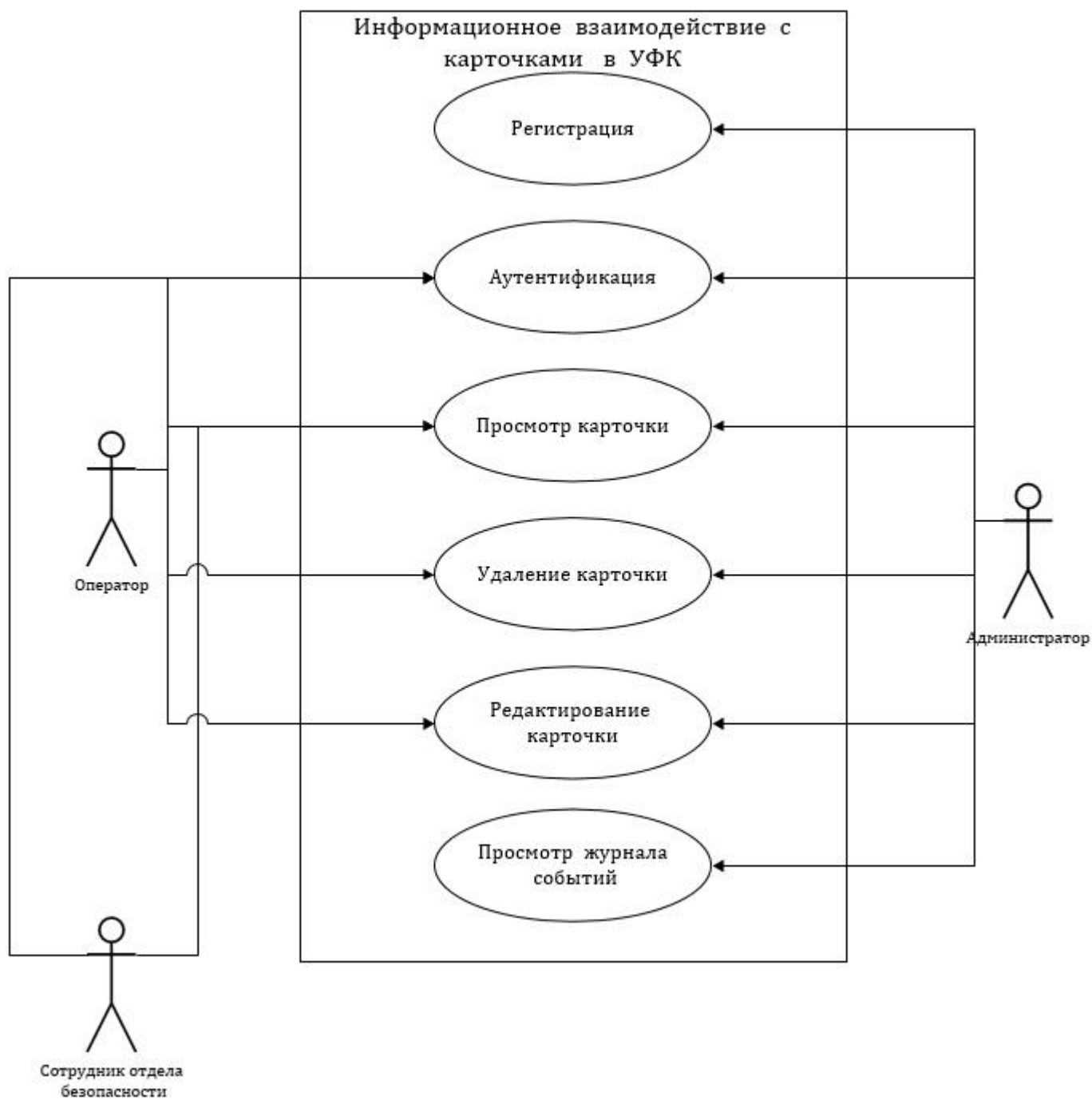


Рисунок 1 – UML Use-case диаграмма

ER диаграмма представлена на рисунке 2.

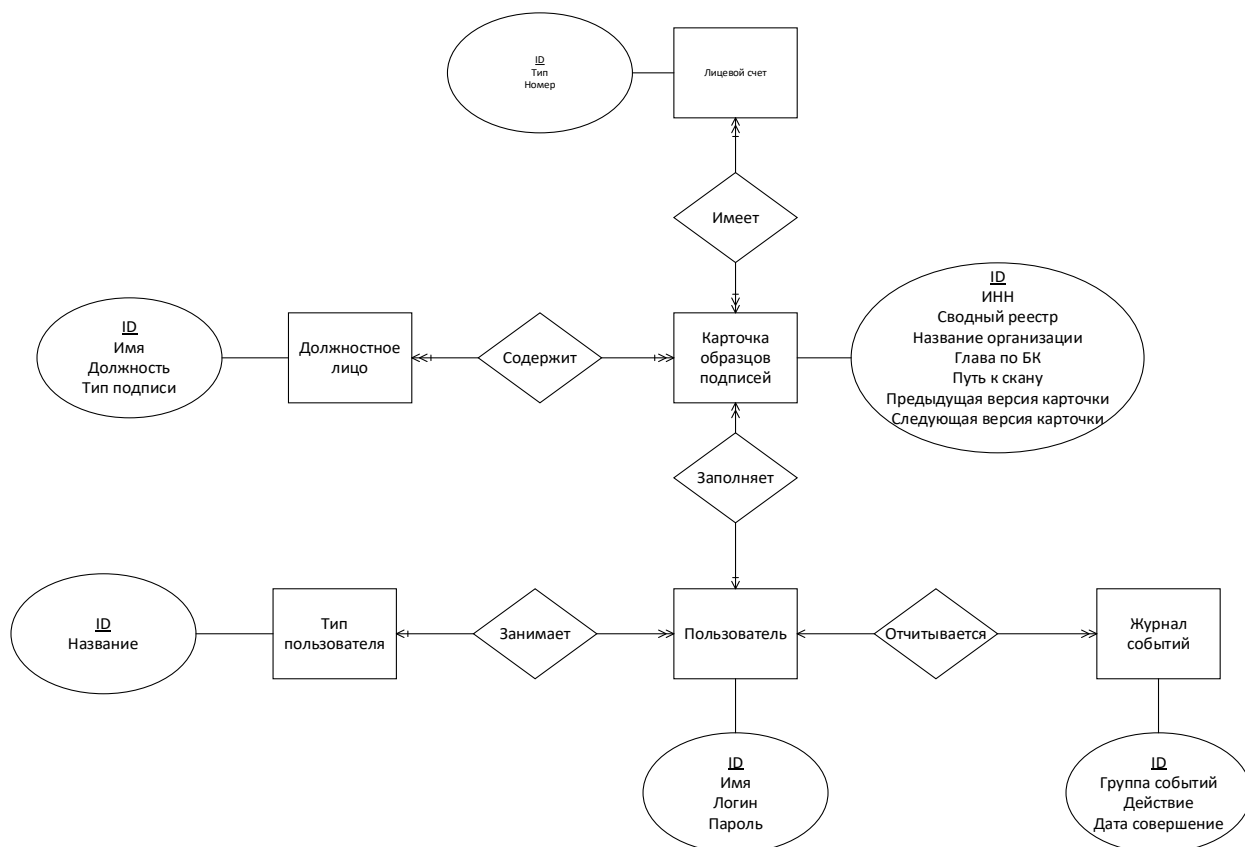


Рисунок 2 – ER диаграмма

2.2 Теоретическое изучение предметной области. Построение теоретических математических моделей

Карточка образцов подписей — это один из обязательных документов, необходимый для оформления открытия лицевого счета соответствующего вида в органе Федерального казначейства.

Требования по формированию Карточки образцов подписей установлены пунктом 4.3 Порядка № 7н. В наименовании формы документа бюджетное учреждение проставляет присвоенный ей номер.

В заголовочной части формы документа клиентом указываются:

1. Дата составления документа.
2. Наименование:
 - а) для участника бюджетного процесса федерального бюджета - кода по Сводному реестру, ИНН и КПП;
 - б) для участника бюджетного процесса субъекта РФ- ИНН и КПП.
3. Полный юридический адрес.

4. Наименование главного распорядителя бюджетных средств.
5. Наименование вышестоящего участника бюджетного процесса.
6. Полное наименование органа Федерального казначейств.

Далее следует раздел «Образцы подписей должностных лиц клиента», разбивается на части:

1. Заполняются полные наименования должностных лиц клиента.

В основном право первой подписи принадлежит руководителю и его заместителям.

Вторую подпись, как правило, закрепляют за главным бухгалтером.

2. Указываются ФИО лиц клиента, которым предоставляются подписи, при совершении операций по лицевому счету.
3. Ставятся образцы подписей должностных лиц.
4. Прописывается срок полномочий каждого должностного лица, временно пользующегося правом подписи.

2.3 Экспериментальное изучение предметной области. Построение эмпирических математических моделей

В ходе разработки была создана диаграмма кооперации – рисунок 3.

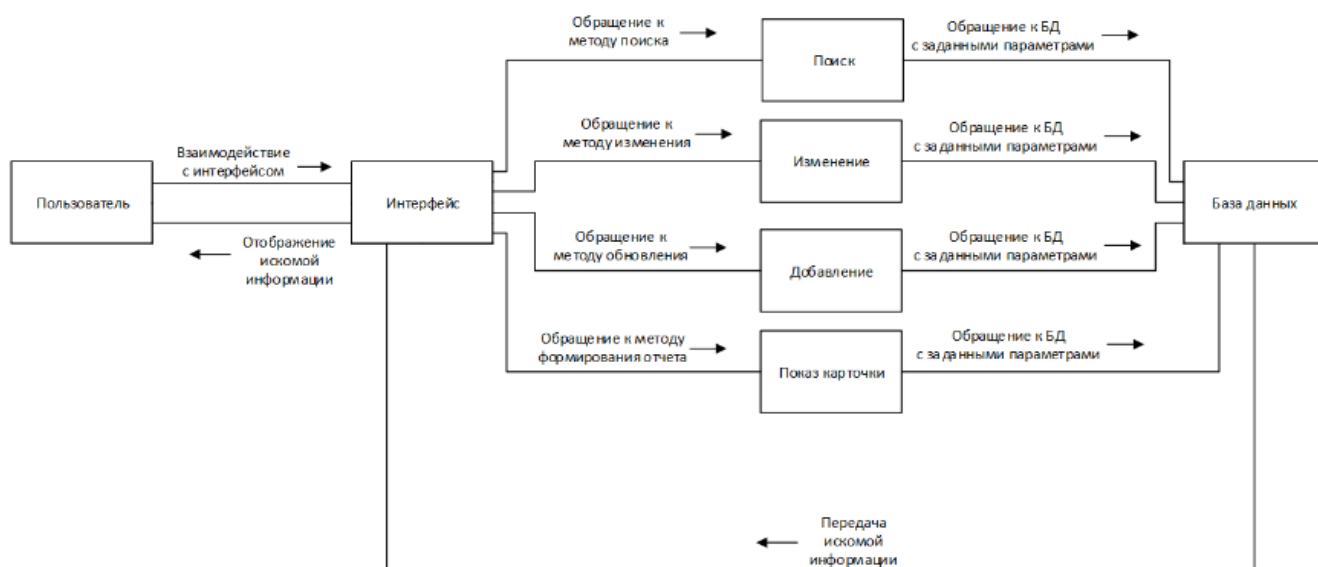


Рисунок 3 – Диаграмма кооперации

Диаграмма состояний системы в разрабатываемой информационной системе представлена на рисунке 4.

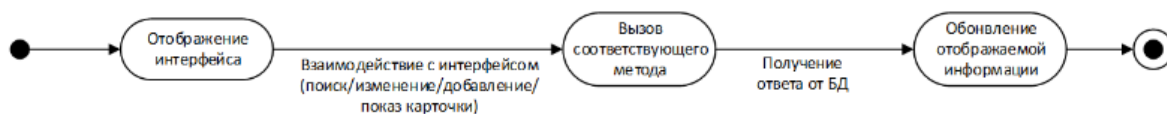


Рисунок 4 – Диаграмма состояний системы

На рисунке 5 представлена диаграмма потоков данных

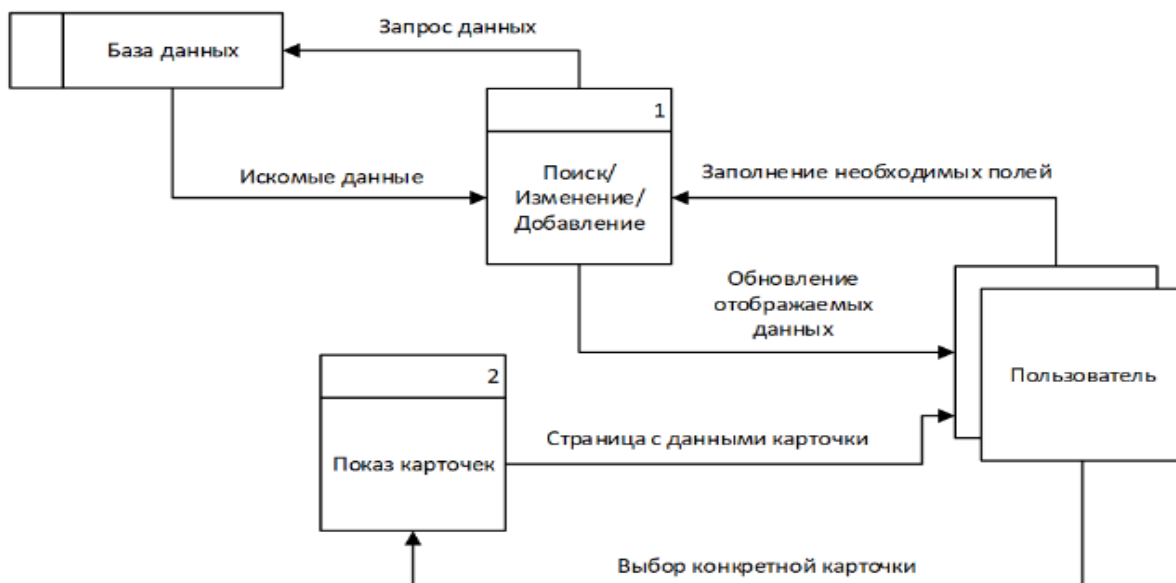


Рисунок 5 – Диаграмма потоков данных

3. Разработка информационной базы для решения задачи

3.1 Построение концептуальной и физической модели данных

В ходе разработки была создана концептуальная модель базы данных показанная на рисунке 6.

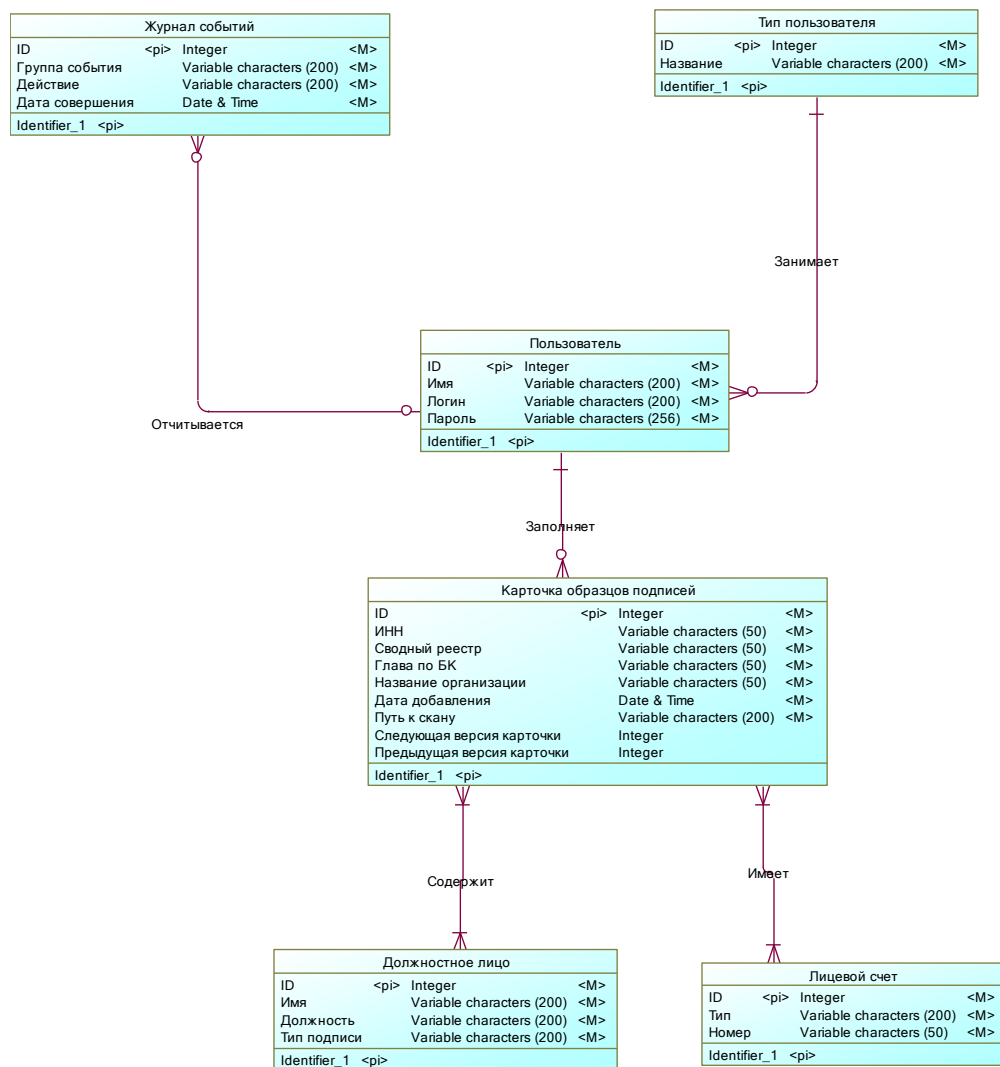


Рисунок 6 – Концептуальная модель базы данных

Спецификация сущностей приведена в таблицах 1 - 6.

Таблица 1 – Спецификация сущности «Журнал событий»

Имя	Код	Тип	Обязательный	Первичный
ID	id	Integer	TRUE	TRUE
Группа события	Name	Variable characters (200)	TRUE	FALSE
Действие	text	Variable characters (200)	TRUE	FALSE
Дата совершения	date	Date & Time	TRUE	FALSE

Таблица 2 – Спецификация сущности «Тип пользователя»

Имя	Код	Тип	Обязательный	Первичный
ID	id	Integer	TRUE	TRUE
Название	name	Variable characters (200)	TRUE	FALSE

Таблица 3 – Спецификация сущности «Пользователь»

Имя	Код	Тип	Обязательный	Первичный
ID	id	Integer	TRUE	TRUE
Имя	name	Variable characters (200)	TRUE	FALSE
Логин	login	Variable characters (200)	TRUE	FALSE
Пароль	password	Variable characters (256)	TRUE	FALSE

Таблица 4 – Спецификация сущности «Карточка»

Имя	Код	Тип	Обязательный	Первичный
ID	id	Integer	TRUE	TRUE
ИНН	inn	Variable characters (50)	TRUE	FALSE
Сводный реестр	Consolidated_registr	Variable characters (50)	TRUE	FALSE
Глава по БК	Head_bk	Variable characters (50)	TRUE	FALSE
Название организации	name	Variable characters (50)	TRUE	FALSE
Дата добавления	date	Date & Time	TRUE	FALSE
Путь к скану	path	Variable characters (200)	TRUE	FALSE
Следующая версия карточки	next_version_id	Integer	FALSE	FALSE
Предыдущая версия карточки	prev_version_id	Integer	FALSE	FALSE

Таблица 5 – Спецификация сущности «Лицевой счет»

Имя	Код	Тип	Обязательный	Первичный
ID	id	Integer	TRUE	TRUE
Тип	type	Variable characters (200)	TRUE	FALSE
Номер	number	Variable characters (50)	TRUE	FALSE

Таблица 6 – Спецификация сущности «Должностное лицо»

Имя	Код	Тип	Обязательный	Первичный
ID	id	Integer	TRUE	TRUE
Имя	name	Variable characters (200)	TRUE	FALSE
Должность	position	Variable characters (200)	TRUE	FALSE
Тип подписи	signature_type	Variable characters (200)	TRUE	FALSE

Спецификация связей приведена в таблицах 7.

Таблица 7 – Спецификация связей

Имя	Код	Сущность 1		Сущность 2	Тип связи
Занимает	Takes	Тип пользователя		Пользователь	One - Many
Отчитывается	Reports	Пользователь		Журнал событий	One - Many
Заполняет	Fills	Пользователь		Карточка образцов подписей	One - Many
Имеет	Have	Карточка образцов подписей	Лицевой счет		Many - Many

Окончание таблицы 7

Имя	Код	Сущность 1	Сущность 2	Тип связи
Содержит	Contains	Карточка образцов подписей	Должностное лицо	Many - Many

После разработки концептуальной модели базы данных была создана физическая модель, представленная на рисунке 7.

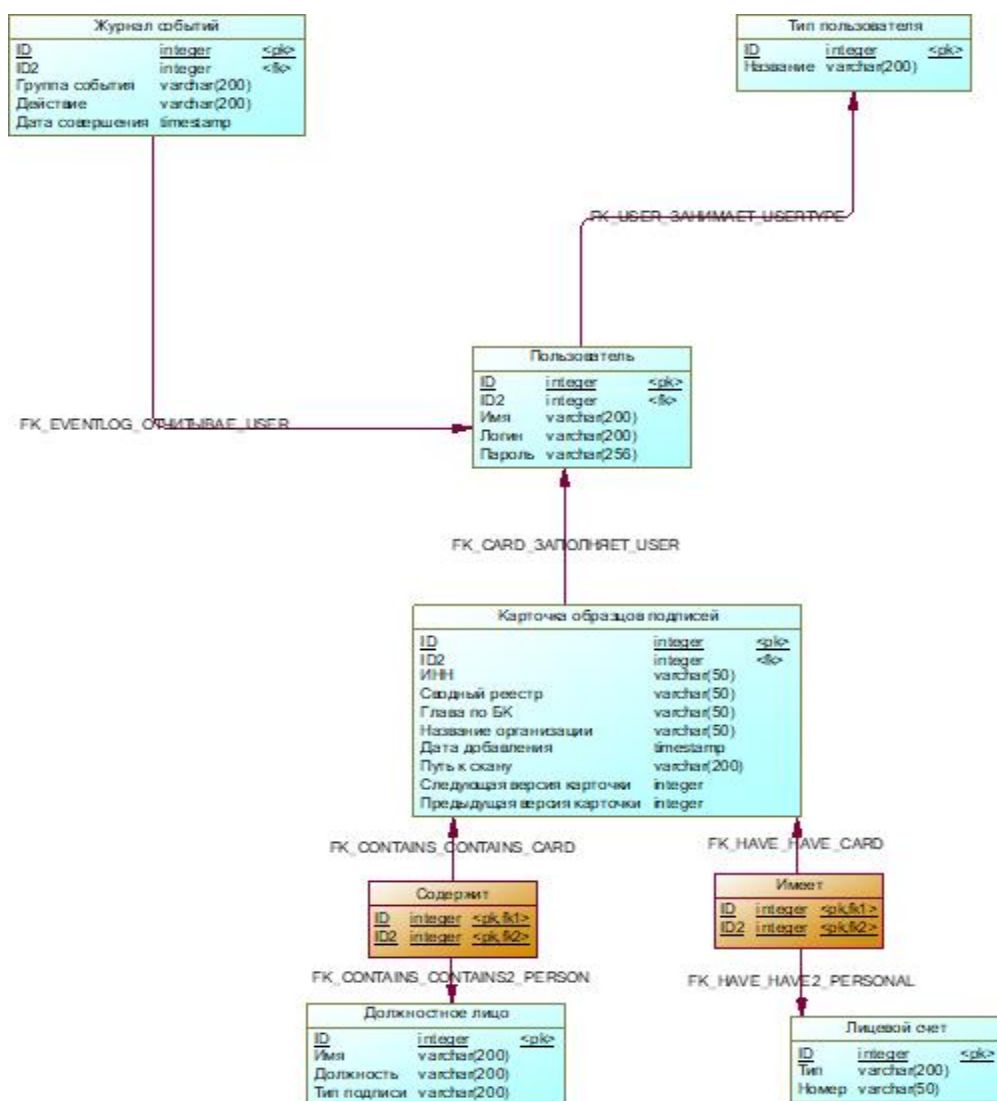


Рисунок 7 – Физическая модель базы данных

Список таблиц базы данных приведен в таблице 8

Таблица 8 – Спецификация таблиц базы данных

Имя	Код
Тип пользователя	UserType
Карточка образцов подписей	Card
Пользователь	User
Должностное лицо	Person
Лицевой счет	PersonalAccount
Журнал событий	EventLog

3.2 Описание источников информации, входных сигналов и документов

Источниками информации разрабатываемой АСУ являются:

1. Входные данные пользователей, полученные от администратора, при их регистрации в системе.
2. Карточки образцов подписей (см. приложение А), которые содержат наименования, номера лицевых счетов, ИНН, сводные реестры, главы по БК организаций, информацию о должностных лиц, имеющих право подписи платежных и иных документов при совершение операций по лицевым счетам, даты добавления и отсканированные версии карточек в формате PDF.

3.3 Описание выходной информации: сигналов, документов и видеок кадров

Выходной информацией в разрабатываемой системе является информация, хранящаяся в карточках образцов подписей. Также есть возможность печати карточек.

4 Программно-аппаратная реализация решения задачи

4.1 Аппаратное обеспечение

Аппаратное обеспечение рассматривается с двух сторон: со стороны сервера и со стороны пользователя.

Системные требования со стороны сервера:

1. Оперативная память: 2 Гбайт.
2. Процессор: Intel Xeon e-2224.
3. Операционная система: Linux Debian.
4. Жесткий диск: 10 Гбайт.

Системные требования для Linux Debian:

1. Минимальное количество памяти 512мб.
2. Минимальное место на диске 2гб.
3. Минимальный поддерживаемый процессор Pentium 4.1Ghz.

Системные требования со стороны пользователя:

1. Необходим доступ к браузеру.
2. Операционная система: Windows 10, 7, 8, 8.1.
3. Разрядность: x86 (32-bit) или x64 (64-bit).
4. Видеоадаптер: 3D адаптер nVidia, Intel, AMD/ATI.
5. Винчестер: 400 Мб.
6. Оперативная память: 512 Мб.
7. Аудиокарта: Любая.
8. Контроллер: Клавиатура, Мышь.
9. Интернет: Стабильное соединение.
10. Разрешение экрана: SVGA 800x600.

4.2 Программное обеспечение

Операционная система Linux Debian

Debian это операционная система с открытым исходным кодом для Linux-серверов, состоящая из свободного ПО. Многие дистрибутивы Linux, основаны на Debian. Debian очень популярен, как ОС для серверов, это связано

с его устойчивостью.

Данная дистрибутив имеет пакетный менеджер, содержащий наибольшее среди других дистрибутивов хранилище пакетов. В Debian входят менеджер пакетов, инструменты АРТ и интерфейсы, такие как GDebi.

Преимущества Debian:

1. Совместимость приложений.
2. Безопасность.
3. Стабильность.

Главная причина использовать Debian это стабильность, а также, он предлагает удобную серверную среду. Debian достаточно прост в настройке.

Для работы с СУБД используется PostgreSQL.

PostgreSQL - это свободно распространяемая объектно-реляционная система управления базами данных (ORDBMS). PostgreSQL очень развитая открытая СУБД. PostgreSQL является альтернативой коммерческих баз данных [3].

PostgreSQL поддерживается на всех Unix системах, включающих в себя 34 платформы, такие как Linux, FreeBSD, NetBSD, OpenBSD, SunOS, Solaris, DUX, а также под Mac OS X.

PostgreSQL является наиболее развитой из СУБД с открытым исходным кодом, которая, может составить конкуренцию коммерческим базам данных.

Преимуществом PostgreSQL считаются:

1. Высокая производительность.
2. Поддержка БД неограниченного размера.
3. Расширяемость.
4. Наследование.
5. Возможность индексации данных в формате JSON.
6. Поддержка многих языков в стандартной поставке при написании триггеров, хранимых процедур.

IDE для python PyCharm.

PyCharm помогает в максимально продуктивной разработке на Python,

JavaScript, CoffeeScript, TypeScript, CSS и популярных языках шаблонов. Включает в себя функции автодополнения, анализа кода, подсветка ошибок и быстрое исправление. Автоматические рефакторинги помогают эффективно редактировать код, а удобная навигация позволяет мгновенно перемещаться по проекту.

Из IDE осуществляется доступ к Oracle, SQL Server, PostgreSQL, MySQL и другим базам данных. А также PyCharm позволяет редактировать SQL-код и выполнять запросы

PyCharm предоставляет большой набор инструментов: встроенный отладчик и инструмент запуска тестов, профилировщик Python, полнофункциональный встроенный терминал, инструменты для работы с базами данных. IDE интегрирована с популярными системами контроля версий, содержит встроенный SSH-терминал, поддерживает возможности удаленной разработки и удаленные интерпретаторы, а также интеграцию с Docker и Vagrant.

PyCharm предоставляет полноценную поддержку разных веб-фреймворков и платформ для разработки на Python, поддерживает темплейтные языки этих фреймворков и JavaScript, CoffeeScript, TypeScript, HTML/CSS, AngularJS, Node.js.

IntelliJ IDEA.

IntelliJ IDEA – это IDE для повышения продуктивности разработки на Java.

IntelliJ IDEA предоставляет возможности для быстрой и эффективной разработки: умное автодополнение, анализ кода в реальном времени и надежные рефакторинги.

IntelliJ IDEA понимает и предоставляет интеллектуальную помощь при написании кода на SQL, JPQL, HTML, JavaScript и многих других языках, а также позволяет редактировать код, написанный не на Java, внутри строковых литералов Java-кода.

Браузер Firefox

Firefox – это один из самых популярных интернет-браузеров. Он удобен в использовании, имеет простой интерфейс, быстро работает и имеет множество возможностей. Доступны версии для компьютеров, работающих на разных операционных системах, а также для смартфонов и планшетов.

В мировом рейтинге Firefox входит в тройку лидеров, уступая по популярности только Google Chrome. Как и его главный конкурент, Firefox отличается безопасностью, широким функционалом, наличием версий для разных типов устройств. Также нельзя не отметить, что программа обладает множеством настроек и дополнений, которые можно загрузить из магазина.

4.3 Разработанные программные средства

Для написания программы использовался язык Python.

Python — это скриптовый язык программирования. Благодаря своей универсальности он подходит для решения разнообразных задач и многих платформ, начиная с iOS и Android и заканчивая серверными ОС.

Python используется в веб-разработке. Самый популярный фреймворк Django. Для создания сайтов на Python используют движки Saleor, Wagtail и т.д. Нередко на Python пишут парсеры.

Не смотря на то что язык не компилируется, на нем создают десктопные программы. Одни из самых знаменитых это GIMP, BitTorrent до 6 версии, Blender.

Также популярна мобильная разработка. Однако чаще всего на Python пишут серверную часть приложения.

Немало компьютерных игр были написаны на Python. Что является прямым доказательством того что этот язык подходит для серьёзных проектов. Python использовался в разработке таких игра, как: Battlefield 2, World of Tanks и Civilization IV.

Несмотря на возможность реализации пользовательского интерфейса и работы с графикой, на Python в основном пишут скрипты — например, взаимодействия персонажей, запуск сцен, а также обработки событий.

Также на Python разрабатываются встроенные системы для устройств.

Python популярен среди системных администраторов. Он необходим для автоматизации задач, так как является достаточно простым, мощным и поддерживает различные спец. пакеты. А также он по умолчанию установлен на все серверы Linux.

В Python есть библиотеки, для проведения исследований и вычислений:

SciPy — библиотека с научными инструментами;

NumPy — расширение, которое добавляет поддержку матриц и многомерных массивов, а также математические функции для работы с ними;

Matplotlib — библиотека для работы с 2D- и 3D-графикой.

Преимущества Python:

1. Это интерпретируемый язык — он не компилируется, то есть до запуска представляет из себя обычный текстовый файл.

2. Быстрая разработка, благодаря уменьшению количества кода, по сравнению с другими языками.

3. Подходит для новичков

Недостатки языка Python:

1. Скорость программ на Python.

2. Сильная зависимость языка от системных библиотек.

Это затрудняет перенос на другие системы.

3. Global Interpreter Lock (GIL) не позволяет выполнять несколько потоков Python одновременно в реализации CPython.

Однако GIL можно отключить на какое-то время, как это сделано в математическом пакете NumPy.

Для клиент серверного приложения используется архитектура SPA.

Одностраничные приложения позволяют имитировать работу десктоп приложений. Архитектура устроена таким образом, что при переходе на новую страницу, обновляется только часть контента. Таким образом, нет необходимости повторно загружать одни и те же элементы. Это очень удобно для разработчиков и пользователей. Для разработки SPA используется один из

самых популярных языков программирования - javascript. Небольшое веб приложение можно сделать с библиотекой jQuery. Вне зависимости какую структуру будущего веб-приложения Вы выберете, уделите внимание графике и пользовательскому интерфейсу.

SPA (single page application) – это веб-приложение, работающее на одной странице. Оно подгружает все необходимые javascript и css файлы при первой загрузке страницы, а затем все общение между клиентом и сервером сводится к минимуму. Т.е. при таком подходе большая часть работы сайта производится на стороне клиента, а если нужно получить данные с сервера, то это обычно делается с помощью JSON.

Такой способ создания сайтов появился относительно недавно, с приходом HTML5, но уже активно набирает обороты. Такое веб-приложение будет работать намного быстрее обычных сайтов, да и разработка не займет много времени. Сейчас уже есть куча фреймворков, которые позволяют создавать очень сложные сайты такого типа достаточно просто и быстро. На данный момент лучшими фреймворками считаются React, Vue.js, Angular. У него больше плюсов, чем у конкурентов, а также он прост в изучении и использовании.

Основные преимущества Single Page Application:

1. Производительность. Так как SPA не обновляет всю страницу, а только нужную часть, это существенно повышает скорость работы.
2. Высокая скорость разработки. Готовые библиотеки и фреймворки дают мощные инструменты для разработки веб приложений. Над проектом могут параллельно работать back-end и front-end разработчики. Благодаря четкому разделению они не будут мешать друг другу.
3. Мобильные приложения. SPA позволяет легко разработать мобильное приложение на основе готового кода.

Недостатки SPA:

1. Плохая SEO оптимизация. SPA работает на основе javascript и загружает информацию по запросу со стороны клиента. Поисковые системы с

трудом могут имитировать данное поведение. Потому большинство страниц попросту недоступны для сканирования поисковыми ботами, поэтому продвижение сайта в 2020 может быть усложнено.

2. Неактивный javascript. Некоторые пользователи отключают javascript в своих браузерах, а без него приложение не будет работать.

Используется Flask – микрофреймворком для создания вебсайтов на языке Python. В основу статьи положен перевод из официальной документации Flask. Поэтому в ней имеется обращение от первого лица, то есть от создателя фреймворка Армина Ронахера.

«Микро» в фреймворке относится не только к простоте и небольшому размеру базы, но это также может означать тот факт, что он не предлагает много проектных решений.

Одним из проектных решений во Flask является то, что простые задачи должны быть простыми; они не должны занимать много кода, и это не должно ограничивать. Flask использует локальные треды внутри объектов, так что пользователь не должен передавать объекты в пределах одного запроса от функции к функции, оставаясь в безопасном тред-объекте. Хотя это и очень простой подход, который позволяет сэкономить время, такое решение может вызывать некоторые проблемы для слишком больших приложений, поскольку изменения в этих локальных тред-объектах могут произойти где угодно в этом тред-объекте.

Во Flask многие вещи предварительно сконфигурированы, на основе общей базовой конфигурации. Например, шаблоны и статические файлы сохранены в подкаталогах в пределах исходного дерева.

Основная причина почему Flask называется «микрофреймворком» — это идея сохранить ядро простым, но расширяемым. В нем нет абстрактного уровня базы данных, нет валидации форм или всего того, что уже есть в других библиотеках. Однако, Flask поддерживает расширения, которые могут добавить необходимую функциональность и имплементируют их так, как будто они уже были встроены изначально. В настоящее время уже есть

расширения: формы валидации, поддержка загрузки файлов, различные технологии аутентификации и многие другие.

Веб-приложение можно взломать различными способами, так как веб-программирование — это небезопасное занятие. Flask защищает приложение от наиболее распространенных и известных способов взлома, такие как XSS (cross-site scripting). До тех пор, пока пользователь сам сознательно не отметит опасный html как безопасный.

В связке с микрофреймворком Flask используется библиотека Flask-SQLAlchemy, которая обеспечивает простой и мощный интерфейс взаимодействия ваших объектов и реляционной базы данных любого типа. Он направлен на упрощение использования SQLAlchemy с Flask, предоставляя полезные значения по умолчанию и дополнительные помощники, которые облегчают выполнение общих задач. В SQLAlchemy взаимодействие с базой данных происходит с помощью сессии.

Также используется библиотека Werkzeug WSGI, которая имеет две стороны: "сервер" или "шлюз" (часто это веб-сервер таких как Apache или Nginx), и сторона "приложение" или "каркас" (сам Python script). Для обработки запроса WSGI серверная сторона выполняет приложение и предоставляет информацию об окружающей среде и обратного вызова на стороне приложения. Процессы приложений запрос, возвращающий ответ на серверную сторону, используя функция обратного вызова была предоставлена.

Между сервером и приложения, может существовать промежуточное ПО WSGI, которое реализует оба сторон API. Сервер получает запрос от клиента и пересылает его в промежуточное ПО. После обработки он отправляет запрос приложение. Ответ приложения отправляется промежуточное ПО на сервер и, в конечном счете, клиенту. Может быть несколько посредников, образующих стек приложений, совместимых с WSGI.

Для снижения риска проведения атак методом инъекции, используется библиотека MarkupSafe.

MarkupSafe реализует текстовый объект, который экранирует символы, так что это безопасный для использования в HTML и XML. Символы, которые имеют специальные значения заменяются таким образом, чтобы они отображались как фактические символы. Благодаря данной библиотеке, пользовательский ввод может быть безопасно обработан сервером.

Для написания современного пользовательского интерфейса используется JavaScript-фреймворк Vue.js с открытым исходным кодом. Vue.js легко интегрируется в проекты с использованием других JavaScript-библиотек. Может функционировать как веб-фреймворк для разработки одностраничных приложений в реактивном стиле.

В 2013 году сотрудник Google Эван, работая над одним из проектов, пришёл к выводу, что не существует готовых решений для быстрого прототипирования сложных пользовательских интерфейсов веб-приложений: React тогда был на ранней стадии разработки, основными инструментами были такие сложные фреймворки, как AngularJS или ориентированный на MVC-архитектуру Backbone.js, не отличавшиеся простотой и ориентированные на разработку больших приложений. Для преодоления этого пробела Ю начал разработку Vue.js, которая, сохраняя простоту, оказалась пригодна не только для прототипирования, но и для полноценной разработки.

В октябре 2015 года была выпущена версия 1.0 библиотеки, в сентябре 2016 года — версия 2.0.

4.3.2 Описание программы

4.3.2.1 Общие сведения

Название программы — «ХКОП». Программа реализована для автоматизации и упрощения процесса заполнения и хранения карточек образцов подписей. В данной работе АИС является web-приложением, созданным на языке Python.

4.3.2.2 Функциональное назначение

Назначением программы является хранение карточек образцов подписей, частичная автоматизация процесса заполнения карточек, учет изменения карточек.

4.3.2.3 Описание логической структуры

Структура программы представляет из себя набор страниц:

1. Страница авторизации.
2. Страница меню «пользователя» (администратора, оператора).
3. Страница пользователей, зарегистрированных в системе.
4. Страница карточек, хранящихся в системе.
5. Страница журнала событий.
6. Страница регистрации новых пользователей.
7. Страница изменения пользователей.
8. Страница каталога карточек.
9. Страница просмотра карточек.
10. Страница изменения карточек.
11. Страница добавления карточек.
12. Страница просмотра истории карточек.
13. Страница поиска.

Взаимодействие модулей друг с другом представлено на рисунке 8.

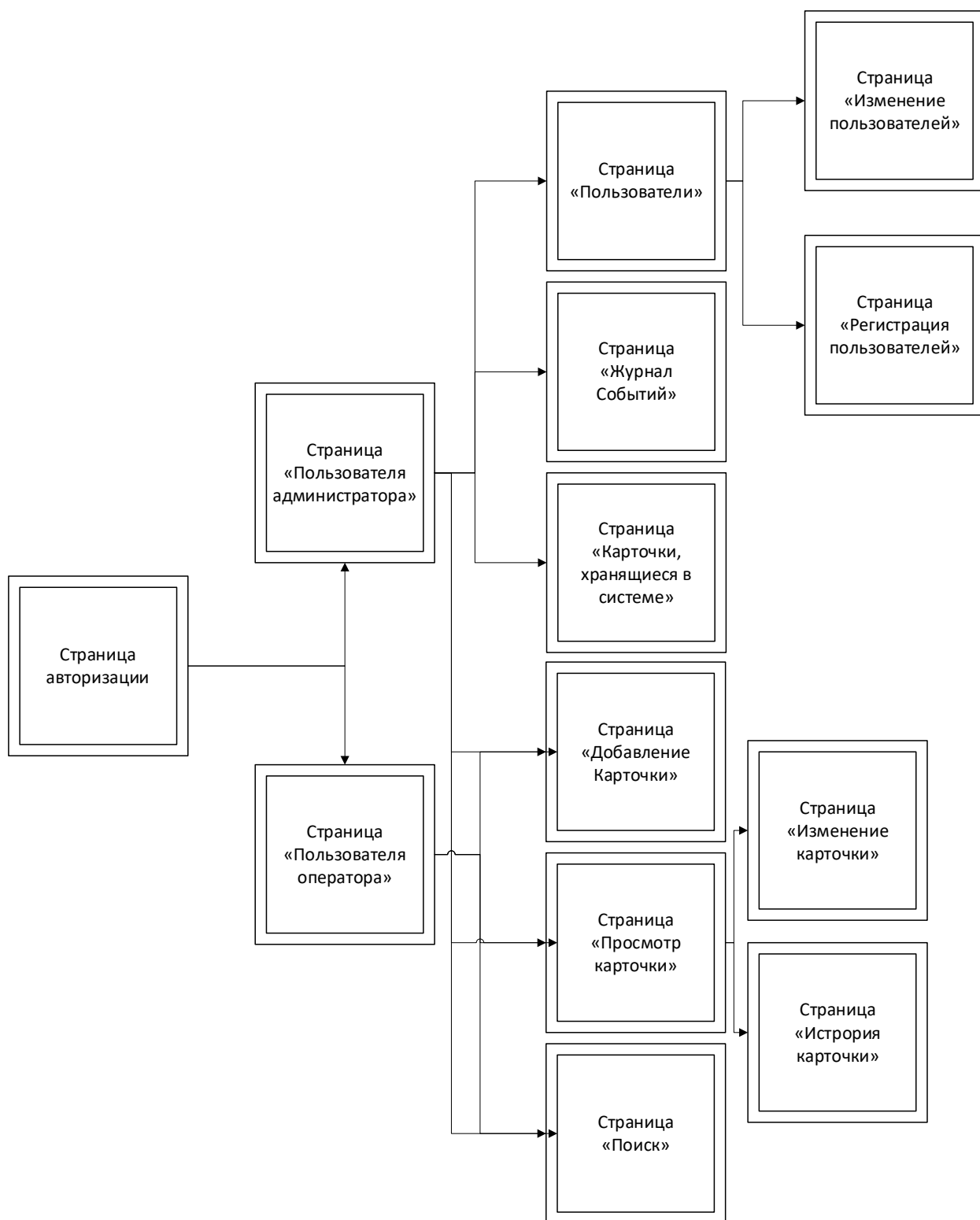


Рисунок 8 – Взаимодействие форм друг с другом

Страница авторизации служит для авторизации пользователя. Страница каталога карточек служит для вывода последних версий карточек с возможностью выбора периода (3 дня, неделя, месяц) и отображаемой информацией (ИНН, сводный реестр, глава по БК, наименование

организации). Страница просмотра карточек служит для отображения информации, хранящейся в карточках образцов подписей (ИНН, сводный реестр, глава по БК, наименование организации, лицевые счета, должностные лица, имеющие право подписи), дата и пользователь, добавивший карточку. Страница просмотра истории изменения карточки служит для вывода информации в карточках предыдущих версий. Страница добавления служит для добавления новой карточки в базу данных. Страница изменения служит для добавления новой карточки в базу данных. Страница поиска служит для облегчения навигации и поиска интересующей информации. Страница просмотра всех карточек доступна только администратору и позволяет вывести все карточки, хранящиеся в базе данных, а также служит для удаления. Страница просмотра всех пользователей доступна только администратору, позволяет вывести всех пользователей, зарегистрированных в системе с возможностью изменения и удаления. Страница просмотра журнала событий позволяет просматривать действия, совершенные пользователем.

4.3.2.4 Используемые технические средства

Программа функционирует на двух системах: серверной и клиентской.

На серверной части:

В работе программы используются компьютеры на базе процессора x86 со стандартной конфигурацией и с установленной операционной системой. Требования к компьютерам обусловлены операционной системой.

Для клиентской части:

Клиентское приложение не зависит от архитектуры процессора и установленной ОС единственное требование поддержка веб браузера последних версий.

4.3.2.5 Вызов и загрузка

Серверная часть программы запускается при помощи команды `flask run`

start.py, запущенная в терминале в директории программы.

Клиентская часть запуск браузера ввод URL приложения.

4.3.2.6 Входные данные

Для администратора:

1. Вносятся личные данные для авторизации.
2. Вносятся данные для создания новой карточки.
3. Вносятся данные для добавления новой версии карточки.
4. Вносятся данные для добавления новых пользователей.
5. Вносятся данные для изменения пользователей.
6. Вносятся данные, требующиеся при использовании функции поиска, при необходимости.

Для оператора:

1. Вносятся личные данные для авторизации.
2. Вносятся данные для создания новой карточки.
3. Вносятся данные для добавления новой версии карточки.
4. Вносятся данные, требующиеся при использовании функции поиска, при необходимости.

4.3.2.7 Выходные данные

Для администратора:

1. Все имеющиеся в системе карточки.
2. Информация, хранящаяся внутри карточек.
3. Журнал событий.
4. Список пользователей (ФИО, Логин, Тип пользователя).

Для оператора:

1. Все имеющиеся в системе карточки.
2. Информация, хранящаяся внутри карточек.

4.3.3 Описание применения программы

4.3.3.1 Назначение программы

Данная программа написана для автоматизации и упрощения процесса хранения карточек образцов подписей. Также она позволяет вести учет изменений карточек.

4.3.3.2 Условия применения

Программа функционирует на двух системах: серверной и клиентской.

На серверной части:

Postgresql версии 12.x.

Интерпретатор Python с библиотеками:

1. Flask.
2. Flask-SQLAlchemy.
3. Werkzeug WSGI.
4. Click.
5. MarkupSafe.

Мышь и клавиатура, монитор по необходимости.

На клиентской части:

Устройство с браузером с поддержкой HTML5 CCS3, а также необходимые периферийные устройства для взаимодействия с интерфейсом приложения.

4.3.3.3 Описание задачи

Основной задачей системы является хранение карточек образцов подписей, частичная автоматизация ввода данных в систему.

4.3.3.4 Входные и выходные данные

Входные данные

Для администратора:

1. Вносятся личные данные для авторизации.
2. Вносятся данные для создания новой карточки.
3. Вносятся данные для добавления новой версии карточки.
4. Вносятся данные для добавления новых пользователей.
5. Вносятся данные для изменения пользователей.
6. Вносятся данные, требующиеся при использовании функции поиска, при необходимости.

Для оператора:

1. Вносятся личные данные для авторизации.
2. Вносятся данные для создания новой карточки.
3. Вносятся данные для добавления новой версии карточки.
4. Вносятся данные, требующиеся при использовании функции поиска, при необходимости.

Выходные данные

Для администратора:

1. Все имеющиеся в системе карточки.
2. Информация, хранящаяся внутри карточек.
3. Журнал событий.
4. Список пользователей (ФИО, Логин, Тип пользователя).

Для оператора:

1. Все имеющиеся в системе карточки.
2. Информация, хранящаяся внутри карточек.

4.3.4 Описание результатов работы программы

Таким образом, программа позволяет повысить эффективность деятельности учреждения, путём частичной автоматизации добавления, хранения и поиска данных, хранящихся в карточках образцов подписей.

5 Результаты внедрения и использования системы. Достижение целей разработки

В ходе работы было разработано приложение для автоматизации процесса хранения карточек образцов подписей.

В приложении реализованы следующие функции:

1. Хранение карточек образцов подписей.
2. Изменение карточек образцов подписей.
3. История карточек образцов подписей.
4. Частичная автоматизация заполнения данных.
5. Поиск карточек образцов подписей.

Главными целями, поставленными для данного проекта, были повышение эффективности работы с карточками образцов подписей. Была усовершенствована система представления, поиска и хранения карточек. Этот функционал был успешно реализован в программе.

В приложении Б представлены скриншоты разработанной программы.

Заключение

Было реализовано приложение, упрощающее работу с карточками образцов подписей, благодаря внедрению систем поиска и хранения.

Приложение реализовано с использованием современной архитектуры. Главными преимуществами использования данной архитектуры являются производительность и низкая зависимость от скорости соединения с интернетом.

Список литературы

1. Выступление Мазеева В.А. [Электронный ресурс]. – Электронные данные. Режим доступа: <http://cuadaro.donland.ru/default.aspx?pageid=102316>.
2. Компания КАДФЕМ Си-Ай-Эс внедрила "Е1 Евфрат" [Электронный ресурс]. – Электронные данные. Режим доступа: <http://avtprom.ru/news/2014/12/17/kompaniya-kadfem-si-ai-e>.
3. PostgreSQL – Википедия. – Электронные данные. Режим доступа: <https://ru.wikipedia.org/wiki/PostgreSQL>

Приложение А. Шаблон карточки образцов подписей.

Приложение № 2
к Порядку открытия и ведения лицевых счетов
территориальными органами Федерального
казначейства, утвержденному приказом Федерального
казначейства
от 17 октября 2016 г. № 21н
(в ред. Приказа Казначейства России от 28.12.2017 №
36н)

КАРТОЧКА ОБРАЗЦОВ ПОДПИСЕЙ № 		Коды	
К ЛИЦЕВЫМ СЧЕТАМ № _____		Форма по КФД	0531753
от "____" _____ 20__ г.		Дата	
Наименование клиента	_____	по ОКПО	
		по Сводному реестру	
		ИНН	
		КПП	
Адрес	_____	Телефон	
Наименование главного распорядителя бюджетных средств, главного администратора источников финансирования дефицита бюджета, главного администратора доходов бюджета	_____	Глава по БК	
Наименование вышестоящего участника бюджетного процесса (вышестоящей организации)	_____	по ОКПО	
		по Сводному реестру	
Орган Федерального казначейства	_____	по КОФК	

Образцы подписей должностных лиц клиента, имеющих право подписи платежных и иных документов при совершении операции по лицевому счету

Право подписи	Должность	Фамилия, имя, отчество	Образец подписи	Срок полномочий лиц, временно пользующихся правом подписи
1	2	3	4	5
первой				
второй				

Руководитель клиента
(уполномоченное лицо)

_____ (должность) _____ (подпись) _____ (расшифровка подписи)

М.П.

Главный бухгалтер клиента
(уполномоченное лицо)

_____ (должность) _____ (подпись) _____ (расшифровка подписи)

Отметка об удостоверении полномочий и подписей

Руководитель

(уполномоченное лицо)

(должность)

(подпись)

(расшифровка подписи)

М.П.

“ ____ ” _____ 20 ____ г.

Удостоверительная надпись о засвидетельствовании подлинности подписей ¹

**Отметка органа Федерального казначейства
о приеме образцов подписей**

Главный бухгалтер

(уполномоченное лицо)

(должность)

(подпись)

(расшифровка подписи)

Ответственный исполнитель

(должность)

(подпись)

(расшифровка подписи)

(телефон)

“ ____ ” _____ 20 ____ г.

Особые отметки _____

¹ При нотариальном заверении заполняется в соответствии со статьей 51 Основ законодательства Российской Федерации о нотариате, утвержденных Верховным Советом Российской Федерации 11 февраля 1993 г. № 4462-1 (Ведомости Съезда народных депутатов Российской Федерации и Верховного Совета Российской Федерации, 1993, № 10, ст. 357; Собрание законодательства Российской Федерации, 2015, № 1, ст. 10; 2016, № 1, ст. 11).

Приложение Б. Интерфейс приложения.

Страница авторизации представлена на рисунке Б.1.

УФК по Липецкой области

Введите логин

Логин

Пароль

Войти

Рисунок Б.1 – Страница авторизации

Страница Каталога карточек представлена на рисунке Б.2.

УФК по Липецкой области

ИНН по СР Глава по ЕК Наименование

Пользователи Карточки Логи

3 дня Неделя Месяц

30.06.2020

1010101052

Добавить

Выйти

Рисунок Б.2 – Страница каталог карточек

Страница всех пользователей, зарегистрированных в системе представлена на рисунке Б.3.

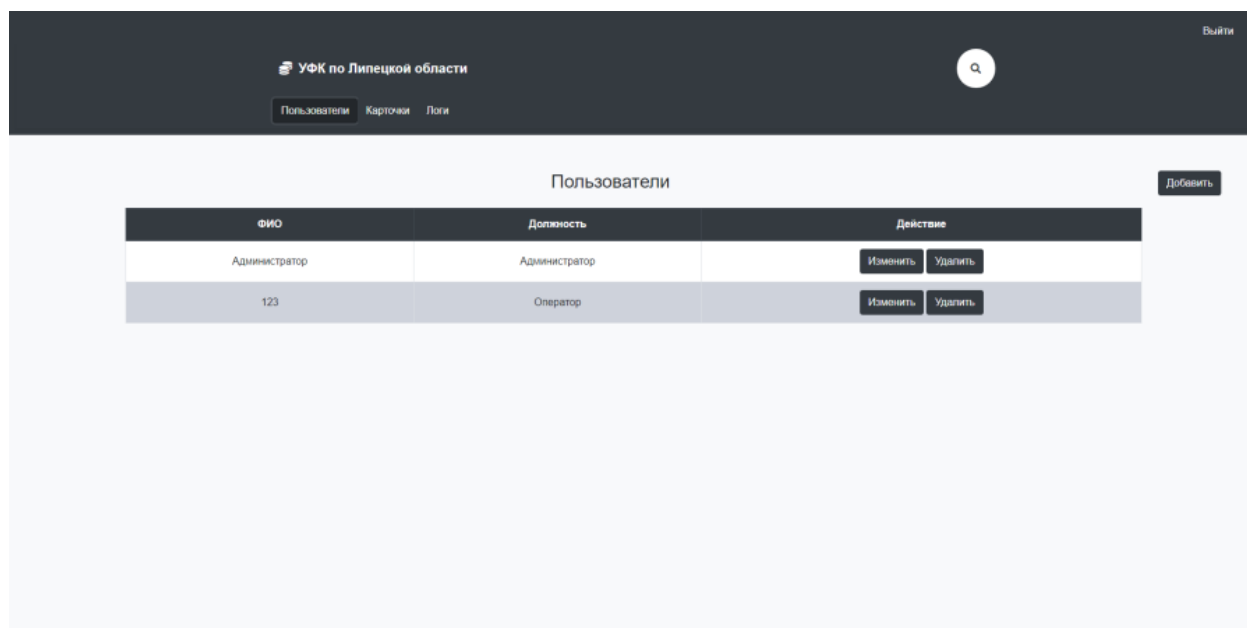


Рисунок Б.3 – Страница всех пользователей

Страница изменения пользователя представлена на рисунке Б.4.

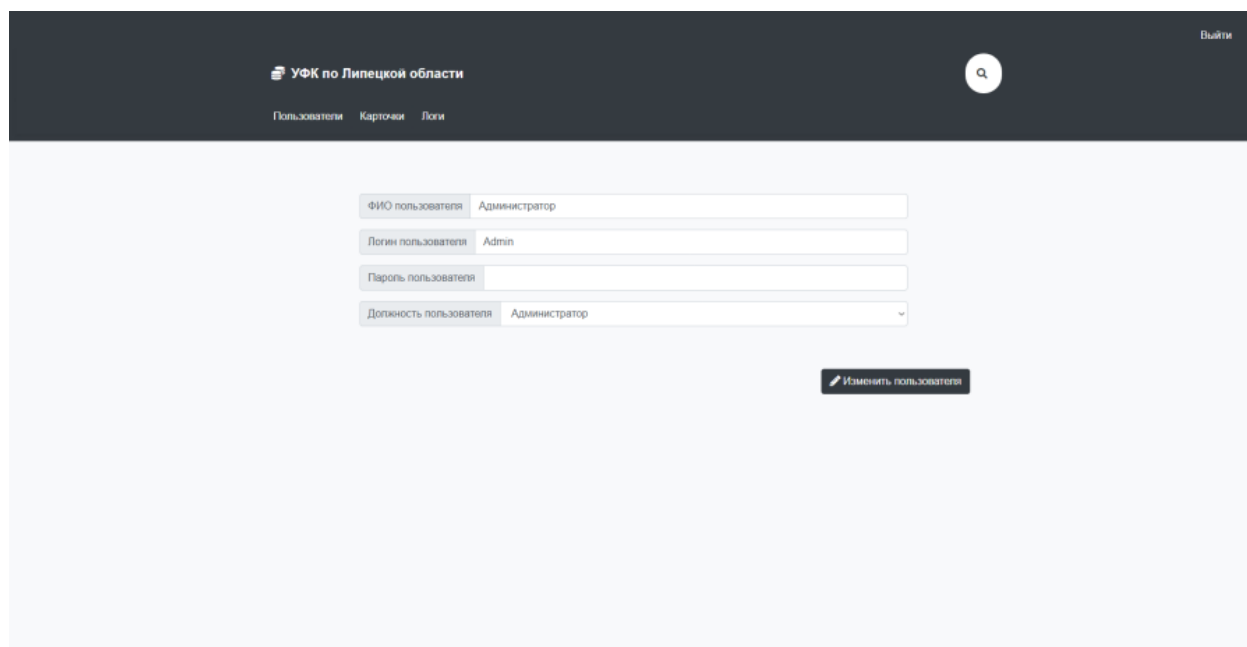


Рисунок Б.4 – Страница изменения пользователя

Страница добавления пользователя представлена на рисунке Б.5.

УФК по Липецкой области

Пользователи Карточки Логи

ФИО пользователя

Логин пользователя

Пароль пользователя

Должность пользователя

Добавить пользователя

Рисунок Б.5 – Страница добавления пользователя

Страница журнала событий представлена на рисунке Б.6.

УФК по Липецкой области

Пользователи Карточки Логи

Журнал событий

Группа события	Текст события	Время действия	Пользователь
Пользователи	Запрос на получение информации о пользователе (1)	30.06.2020, 18:37	Администратор
Пользователи	Запрос на получение пользователей	30.06.2020, 18:37	Администратор
Пользователи	Запрос на получение информации о пользователе (1)	30.06.2020, 18:37	Администратор
Пользователи	Запрос на получение пользователей	30.06.2020, 18:37	Администратор
Карточки	Запрос на получение всех карточек	30.06.2020, 18:37	Администратор
Пользователи	Запрос на получение пользователей	30.06.2020, 18:36	Администратор
Карточки	Запрос на получение всех карточек	30.06.2020, 18:36	Администратор
Карточки	Запрос на получение всех карточек	30.06.2020, 18:36	Администратор
Карточки	Изменение карточки (1)	30.06.2020, 18:36	Администратор
Карточки	Запрос на получение карточки (1)	30.06.2020, 18:32	Администратор
Карточки	Запрос на получение карточки (1)	30.06.2020, 18:32	Администратор
Карточки	Запрос на получение всех карточек	30.06.2020, 18:32	Администратор

Рисунок Б.6 – Страница журнала событий

Страница карточек, хранящихся в системе представлена на рисунке Б.7.

УФК по Липецкой области								
Пользователи Карточки Логи								
Все карточки								
ИНН	Сводный реестр	Глава по БК	Наименование организации	Лицевые счета	Должностные лица	Должность	Тип подписи	Действие
1010101052	12659876532541254896	034	Липецк-Тест	Основной - 1231231230	Ведичев Виталий Викторович	Директор	Первичная	
				Вторичный - 3213213210	Иванов Василий Иванович	Заместитель директора	Вторичная	Удалить
					Иванова Василиса Петровна	Бухгалтер	Первичная	
1231230	1231230	1231230	Липецк-Тест	Основной - 1231231230	Иванов Иван Иванович	Директор	Первичная	
				Вторичный - 3213213210	Иванов Василий Иванович	Заместитель директора	Вторичная	Удалить
					Иванова Василиса Петровна	Бухгалтер	Первичная	

Рисунок Б.7 – Страница карточек, хранящихся в системе

Страница просмотра карточки представлена на рисунке Б.8

УФК по Липецкой области

Пользователи Карточки Логи

Администратор

30.06.2020

Посмотреть историю карточек

Липецк-Тест

ИНН 1010101052

Сводный реестр 12659876532541254896

Глава по БК 034

Лицевые счета

Должностные лица

Тип подписи/должность/ФИО

Первичная

Директор

Ведичев Виталий Викторович

Вторичная

Заместитель директора

Иванов Василий Иванович

Первичная

Бухгалтер

Иванова Василиса Петровна

Открыть pdf

Изменить

Рисунок Б.8 – Страница просмотра карточки

Страница истории карточки представлена на рисунке Б.9

УФК по Липецкой области

Выйти

Пользователи

Карточки

Логи

Q

История карточки

ИНН	Сводный реестр	Глава по БК	Наименование организации	Лицевые счета	Должностные лица	Должность	Тип подписи
1010101052	12659876532541254896	034	Липец-Тест	Основной - 1231231230	Ведищев Виталий Викторович	Директор	Первичная
				Вторичный - 3213213210	Иванов Василий Иванович	Заместитель директора	Вторичная
					Иванова Василиса Петровна	Бухгалтер	Первичная
1231230	1231230	1231230	Липец-Тест	Основной - 1231231230	Иванов Иван Иванович	Директор	Первичная
				Вторичный - 3213213210	Иванов Василий Иванович	Заместитель директора	Вторичная
					Иванова Василиса Петровна	Бухгалтер	Первичная

Рисунок Б.9 – Страница истории карточки

Страница добавления карточки представлена на рисунке Б.10

УФК по Липецкой области

Выйти

Пользователи

Карточки

Логи

Q

Наименование организации

ИНН

Сводный реестр

Глава по БК

Дата добавления

Загрузить PDF

Лицевые счета

Должностные лица

dd

mm

yyyy

Добавить

Рисунок Б.10 – Страница добавления карточки

Страница изменения карточки представлена на рисунке Б.11.

УФК по Липецкой области

Выйти

Пользователи

Карточки

Логин

Наименование организации

Липецк-Тест

ИНН

1010101052

Сводный реестр

12659876532541254896

Глава по БК

034

Дата добавления

30.06.2020

Загрузить PDF

Лицевые счета

Тип счета/номер счета

Основной

1231231230

Вторичный

3213213210

Должностные лица

Добавить лицевой счет

Изменить

Рисунок Б.11 – Страница изменения карточки

Страница поиска представлена на рисунке Б.12.

УФК по Липецкой области

Выйти

Пользователи

Карточки

Логин

Результат по запросу "ИНН = 1231230"

ИНН	Сводный реестр	Глава по БК	Наименование организации	Лицевые счета	Должностные лица	Должность	Тип подписи
1231230	1231230	1231230	Липецк-Тест	Основной - 1231231230	Иванов Иван Иванович	Директор	Первичная
				Вторичный - 3213213210	Иванов Василий Иванович	Заместитель директора	Вторичная
					Иванова Василиса Петровна	Бухгалтер	Первичная

Рисунок Б.12 – Страница поиска