МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»

**ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И   
ТЕХНОЛОГИЧЕСКОГО ОБРАЗОВАНИЯ**

**Кафедра информационных технологий и электронного обучения**

Основная профессиональная образовательная программа

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) «Технологии разработки программного обеспечения»

форма обучения – очная

**Курсовая работа**

«Методология управления разработкой системы электронного документооборота»

Обучающегося 3 курса

Мозгового Никиты Александровича

Научный руководитель:

Кандидат физико-математических наук,   
доцент кафедры ИТиЭО

Жуков Николай Николаевич

Санкт-Петербург

2021

ОГЛАВЛЕНИЕ

[**ВВЕДЕНИЕ** 3](#_Toc93054081)

[ГЛАВА 1. ТЕОРЕТИЧЕСКИЙ АНАЛИЗ 4](#_Toc93054082)

[1.1. Анализ проблемы 4](#_Toc93054083)

[1.2. Обзор существующих решений проблемы 5](#_Toc93054084)

[1.3. Анализ существующих решений 8](#_Toc93054085)

[1.4. Анализ языковых средств для разработки 9](#_Toc93054086)

[1.5 Теория по управлению программным проектом 9](#_Toc93054087)

[ГЛАВА 2. УПРАВЛЕНИЕ РАЗРАБОТКОЙ 13](#_Toc93054088)

[2.1 Техническое задание 13](#_Toc93054089)

[2.2 Этапы разработки frontend составляющей 14](#_Toc93054090)

[2.3 Этапы разработки backend составляющей 16](#_Toc93054091)

[2.4. UML-диаграммы 17](#_Toc93054092)

[ЗАКЛЮЧЕНИЕ 19](#_Toc93054093)

[ЛИТЕРАТУРА 20](#_Toc93054094)

[ПРИЛОЖЕНИЕ А 21](#_Toc93054095)

[ПРИЛОЖЕНИЕ Б 23](#_Toc93054096)

**ВВЕДЕНИЕ**

Документооборот - один из важнейших процессов в работе любой крупной организации. Данный процесс подразумевает возможность добавления, изменения, удаления, своевременного доступа к достоверной информации сотрудником этой организации, причем с определенным разграничением прав для каждого сотрудника. Система для автоматизации этого процесса планируется для разработки и внедрения для “Главного управления МЧС г. Санкт-Петербурга”.

Актуальность работы обусловлена наличием заказа со стороны “Главного управления МЧС города Санкт-Петербурга”.

В данной работе будет описан процесс разработки информационной системы документооборота.

Цель: разработать систему электронного документооборота.

Объект: процесс документооборота в МЧС

Предмет: цифровизация документооборота

Задачи, поставленные для достижения данной цели:

1. Изучить документооборот организации по предоставленному макету
2. Разработать модель базы данных для отображения взаимосвязи сущностей, участвующих в процессе.
3. Разработать серверную часть системы документооборота.
4. Разработать клиентскую часть системы документооборота.
5. Объединить части в единое приложение

# ГЛАВА 1. ТЕОРЕТИЧЕСКИЙ АНАЛИЗ

## 1.1. Анализ проблемы

Документооборот — деятельность по организации движения документов на предприятии с момента их создания или получения до завершения исполнения: отправки из организации и (или) направления в архив.

Документы, поступающие в организацию, проходят:

1. первичную обработку;
2. предварительное рассмотрение;
3. регистрацию;
4. рассмотрение руководством;
5. передачу на исполнение

В случае бумажного документооборота вышеописанные функции распределяются между сотрудниками. Самыми трудозатратными из них чаще всего являются хранение и архивирование: если документ хранился в одном месте у сотрудника определенное время и в дальнейшем был перенесен в архив, об этом нужна соответствующая запись в соответствующем журнале. Ведь только так можно обеспечить возможность поиска необходимых документов. Это усложняет процесс учета и поиска документов [3].

Электронный же документооборот лишен данного минуса. Он позволяет автоматизировать и централизовать процесс архивирования и хранения документов: все документы хранятся в электронном виде, процесс архивирования можно настроить автоматически исходя из типа документа и даты его внесения. Доступ к документам также настраивается: легко настроить распределенный доступ при котором соответствующим пользователям доступны только соответствующие документы, доступ возможен из разных мест благодаря возможностям сетей передачи данных.

Вышеописанное делает электронный документооборот более приоритетным при выборе типа документооборота для организации [4].

На предприятии-заказчике на данный момент действует бумажный документооборот. Конкретно в нашем случае рассматриваются следующие процессы:

* учет аппаратов оснащения в ПСЧ - в т.ч. аппараты, баллоны и любое оборудование / инструменты, записанные за ПСЧ;
* учет укомплектованности ПА и АГ;
* учет учебных занятий;
* учет включений (выездов).

Для организации вышеперечисленных процессов необходимо также организовать следующее:

* учет личного состава;
* учет ПА и АГ;
* учет ПСЧ и ГДЗС - в каждой ГДЗС может быть несколько ПСЧ.

Примеры процесса документооборота:

* после включения по прибытию в ПСЧ ответственный сотрудник заполняет информацию о включении, в т.ч. проводится учет использованного оснащения и заполнение всех баллонов с последующим документированием на бумаге;
* каждый год проходит переучет баллонов и аппаратов, проверяется срок годности, дата освидетельствования каждой единицы и составляется сводная таблица по всем ПСЧ.

При бумажном документообороте не исключены частые ошибки в данных примерах, в особенности в составлении отчетов при переучете. Электронная система документооборота поможет облегчить эту задачу и в случае ошибки или неправильно поданных данных найти источник проблемы.

## 1.2. Обзор существующих решений проблемы

Существует определенное количество готовых решений для автоматизации документооборота на предприятии. Рассмотрим некоторые из них:

1С:Документооборот

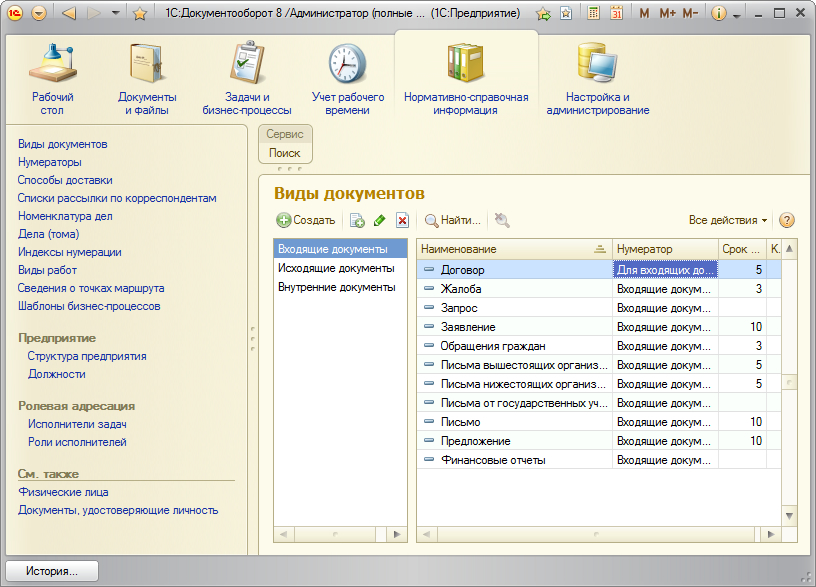


Рисунок 1 - интерфейс 1С:Документооборот

«1С:Документооборот 8» позволяет:

* Упорядочить работу сотрудников с документами, исключить возможность утери версий или пересечения фрагментов при одновременной работе.
* Сократить время поиска нужной информации и суммарное время коллективной обработки документов.
* Повысить качество готового материала (проектов, документации и пр.) за счет решения большого количества спорных вопросов и упорядочивания работы пользователей.

«1С:Документооборот 8» в комплексе решает задачи автоматизации учета документов, взаимодействия сотрудников, контроля и анализа исполнительской дисциплины:

* Централизованное безопасное хранение документов.
* Оперативный доступ к документам с учетом прав пользователей.
* Регистрация входящих и исходящих документов.
* Просмотр и редактирование документов.
* Контроль версий документов.
* Полнотекстовый поиск документов по их содержанию.
* Работа с документами любых типов:
* офисными документами;
* текстами;
* изображениями;
* аудио- и видеофайлами;
* документами систем проектирования;
* архивами;
* приложениями и т.д.
* Коллективная работа пользователей с возможностью согласования, утверждения и контроля исполнения документов.
* Маршрутизация документов, настраиваемая по каждому виду документов в отдельности.
* Автоматизированная загрузка документов из электронной почты и со сканера.
* Учет и контроль рабочего времени сотрудников.

Е1 Евфрат

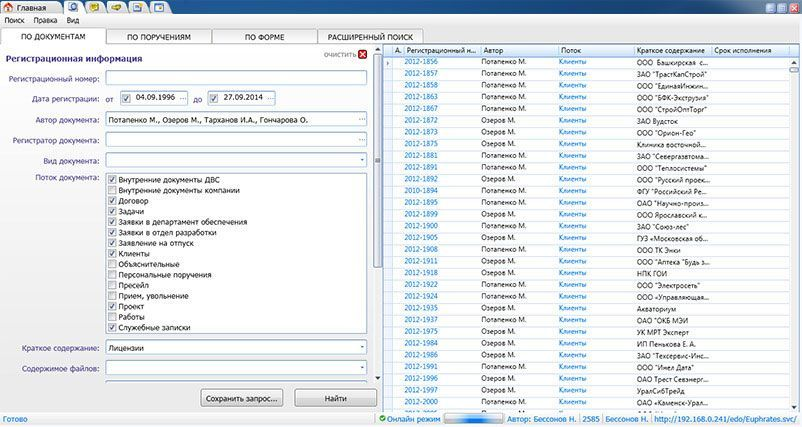


Рисунок 2 - интерфейс Е1 Евфрат

Система распространяется по подписке. В Е1 реализован весь необходимый функционал для организации электронного архива (в том числе и в соответствии с требованиями ГОСТ). 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.

Контур.Диадок

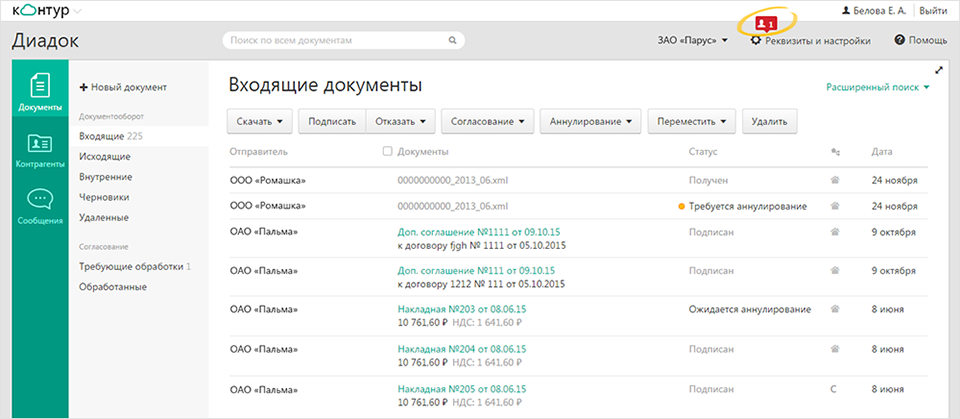


Рисунок 3 - интерфейс Контур.Диадок

Данная система распространяется по подписке. Диадок – это система юридически значимого электронного документооборота, с помощью которой организации обмениваются документами без дублирования на бумаге. Сервис позволяет настроить электронный документооборот внутри организации, с контрагентами и контролирующими органами. Документы передаются через интернет и хранятся в электронных архивах на защищенных серверах.

Через ЭДО передаются:

- декларации НДС;

- счета-фактуры;

- акты выполненных работ, товарные накладные ТОРГ-12;

- договоры, платежные поручения, письма и т. д.

## 1.3. Анализ существующих решений

В ходе изучения доступных решений был сделан вывод, что все современные решения систем электронного документооборота представляют собой хоть и настраиваемую, но достаточно сложную систему.

Для их внедрения необходимо не только тратить человеческие ресурсы для установки и настройки самой системы, но также провести обучение каждого сотрудника. Также такие системы зачастую требовательны к аппаратной части, что усложняет работу сотрудников.

Специфика же работы заказчика предполагает частые выезды, а также работу с документами не только на рабочем месте.

Учитывая все вышеперечисленные факторы нами было принято решение о целесообразности разработки собственной системы, которая будет адаптирована к требованиям заказчика.

## 1.4. Анализ языковых средств для разработки

Вследствие удобства и популярности командой был выбран Django, популярный фреймворк для разработки веб приложений на языке Python. Django имеет ряд отличительных особенностей по умолчанию:

* удобная панель администратора для управления сайтом;
* готовые функции и классы для отображения, создания и изменения данных;
* аутентификация пользователя;
* безопасность, например, от SQL инъекций.

Несмотря на то, что Django является backend-фреймворком, он может использоваться и при разработке пользовательского интерфейса с помощью встроенного щаблонизатора Jinja и применения CSS.

Сейчас стандартной практикой является активное обновление DOM-дерева без перезагрузки страницы, но за отсутствием опыта в React и ему подобных обойдемся перезагрузкой страницы для обновления отображаемой информации.

## 1.5 Теория по управлению программным проектом

Проект - это уникальный набор процессов, состоящий из скоординированных и управляемых задач с начальной и конечной датами, предпринятых для достижения цели.

Управление разработкой программного проекта - это особый вид управления проектами, в рамках которого происходит планирование, отслеживание и контроль за проектами по разработке программного обеспечения. Ключевым моментом в управлении проектом по разработке программного обеспечения является правильный выбор метода разработки.

Процесс разработки программного обеспечения — процесс, посредством которого потребности пользователей преобразуются в программный продукт. Методология разработки - это набор методов по управлению разработкой ПО, набор практических правил как нужно разрабатывать ПО. Модель разработки программного обеспечения описывает, какие стадии жизненного цикла оно проходит и что происходит на каждой из них.

Модели процесса

* Формирование требований;
* Проектирование;
* Реализация;
* Тестирование;
* Внедрение;
* Эксплуатация и сопровождение.

Существует множество различных методологий и моделей разработки. Однако на данный момент в IT-индустрии преобладают гибкие методологии разработки, такие как Scrum и Kanban.

SCRUM

Над каждым проектом работает универсальная команда специалистов, в которой нет внутренней иерархии: ни руководителей, ни подчиненных, ни указаний-приказов. К команде присоединяется еще два человека: владелец продукта и scrum-мастер. Первый соединяет команду с заказчиком и следит за развитием проекта. Второй помогает первому организовать бизнес-процесс: проводит общие собрания, решает бытовые проблемы, мотивирует команду и следит за соблюдением scrum-подхода.

Scrum-подход делит рабочий процесс на равные спринты – обычно это периоды от недели до месяца, в зависимости от проекта и команды. Перед спринтом формулируются задачи на данный спринт, в конце – обсуждаются результаты, а команда начинает новый спринт. Спринты очень удобно сравнивать между собой, что позволяет управлять эффективностью работы.

Преимуществами данного метода является скорость запуска проекта, даже с минимальным бюджетом, а также возможность использования продукта, полученного после окончания каждого спринта. На каждом отдельном этапе выходит готовый продукт и с каждым новым спринтом этот продукт меняется, что позволяет быстро подстраиваться под изменения внешней среды. Ежедневный контроль над ходом работ позволяет более гибко управлять бюджетом проекта. Благодаря участию заказчика продукта в процессе создания эти изменения проходят наиболее эффективно.

KANBAN

Kanban – метод управления разработкой, способствующий равномерному распределению нагрузки между работниками. При данном подходе весь процесс разработки прозрачен для всех членов команды. Задачи по мере поступления заносятся в отдельный список, откуда каждый разработчик может извлечь требуемую задачу.

Вся команда едина – в канбан нет ролей владельца продукта и scrum-мастера. Бизнес-процесс делится на стадии выполнения конкретных задач: «Планируется», «Разрабатывается», «Тестируется», «Завершено» и др.

Главный показатель эффективности в канбан – это среднее время прохождения задачи по доске. Задача прошла быстро – команда работала продуктивно и слаженно. Задача затянулась – надо думать, на каком этапе и почему возникли задержки и чью работу надо оптимизировать.

Для визуализации используют доски: физические и электронные. Они позволяют сделать рабочий процесс открытым и понятным для всех специалистов, что важно, когда у команды нет одного формального руководителя.

Для этого проекта была выбрана первая методология по следующим причинам:

* Во-первых, имелись строгие временные рамки выполнения проекта: необходимо было успеть реализовать продукт к дате защиты курсовой работы. Kanban никаких временных ограничений не имеет, что мешало бы организации процесса.
* Во-вторых, реализуемое приложения представляет из себя единое целое и для должной коммуникации команды и координированного внесения изменений необходимо с определенной частотой проводить встречи для подведения итогов на текущий момент и решения о выпуске продукта. В итоге финальная версия проекта была утверждена ровно в срок, после проведения нескольких спринтов, на которых утверждались или отклонялись внесенные изменения.
* В-третьих, наличие ролей. В методологии Kanban у всех участников команды равные права и ответственность. В нашем случае из-за разницы в опыте и умениях, требовался Scrum-мастер, которым выступил я. В мои обязанности входила инициация проведения спринтов, их обсуждения и формировании на их основании задач и сроков.

# ГЛАВА 2. УПРАВЛЕНИЕ РАЗРАБОТКОЙ

В качестве инструмента для управления проектом был выбран YouTrack от JetBrains. На протяжении всего процесса разработки добавлялись и перемещались задачи.

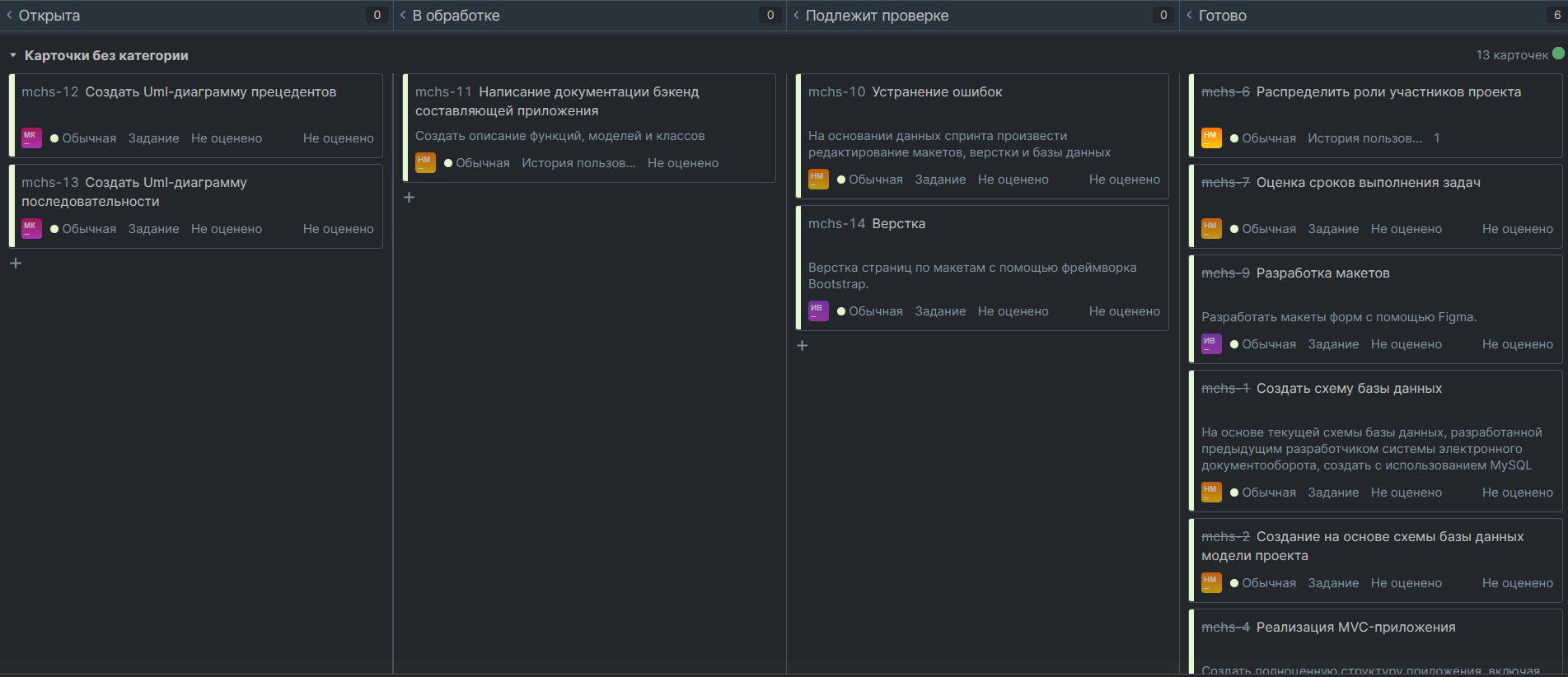


Рисунок 4 - фрагмент рабочей доски проекта.



Рисунок 5 - диаграмма Ганта разработки backend’а

## 2.1 Техническое задание

**Назначение разработки / Общие сведения**

Проект является веб-приложением, представляющим собой систему электронного документооборота для сотрудников МЧС.

**Требования к программной совместимости**

Веб-приложение должно корректно работать в различных браузерах на ПК.

**Требования к содержимому сайта**

Веб-приложение должно содержать следующие страницы:

* Страница регистрации и аутентификации
* Страница с данными пользователей
* Страница профиля работника
* Страница редактирования профиля

**Требования к функциональности приложения**

Система должна позволять пользователю следующие манипуляции:

1. Производить регистрацию и аутентификацию
2. Отображать данные о пользователях
3. Отображать отдельно данные текущего пользователя
4. Заполнять и изменять данные пользователя

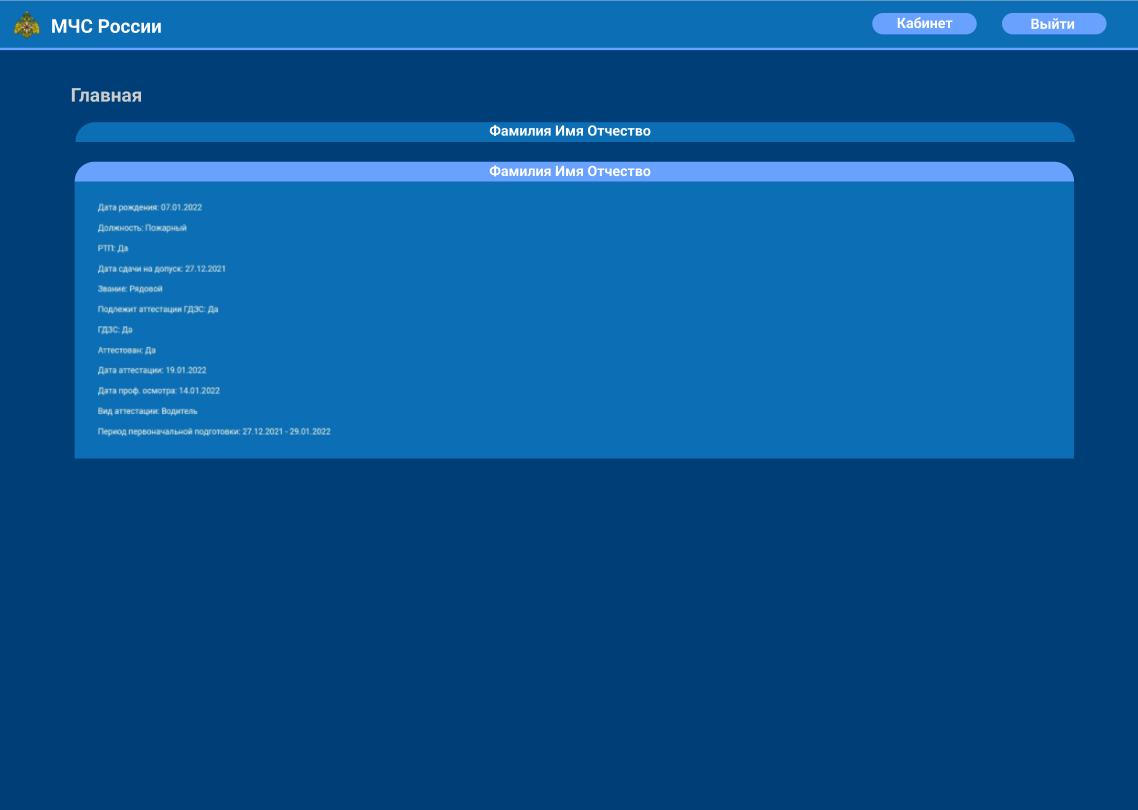
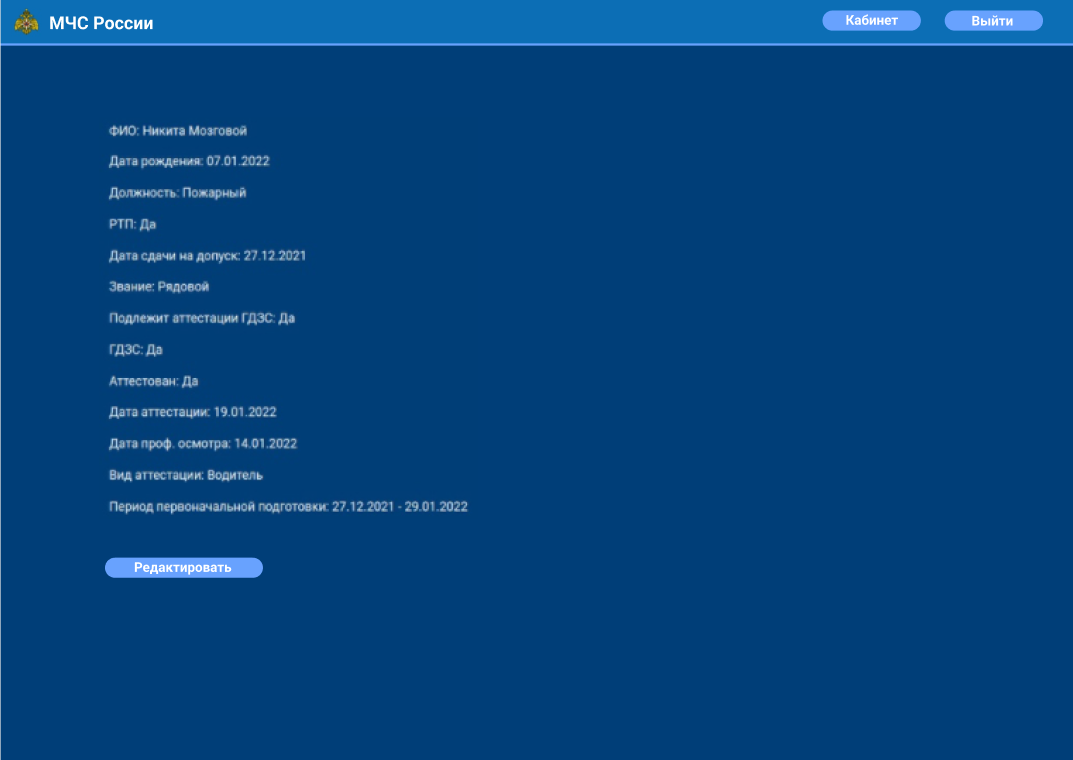
**Требования к дизайну**

Дизайн должен соответствовать таковому на имеющемся сайте МЧС.

**Требования к информационной и программной совместимости**

Приложение должно быть запущено на сервере под управлением ОС семейства Linux (Debian или Ubuntu) или Microsoft Server.

## 2.2 Этапы разработки frontend составляющей

С помощью Figma дизайнером команды были составлены макеты каждой из страниц. Далее они были сверстаны на HTML с использованием CSS, JS и Jinja. Рисунок 6 – макет главной страницыРисунок 7 – макет страницы профиля

Остальные макеты прилагаются к тексту курсовой работы.

## 2.3 Этапы разработки backend составляющей

Прежде всего была продумана модель базы данных, основанная на примерном макете потока данных от заказчика.

Модели проекта внедрялись поэтапно, начиная с модели пользователя, по которой проводится аутентификация и которая содержит общие данные о сотруднике. Новая модель создавалась только после полной проверки на работоспособность в представлениях и контроллерах предыдущих, дабы избежать наследования ошибок. После создания всех моделей производилось их связывание в формате «один ко многим» с помощью внешних ключей. В итоге визуальное представление базы данных таково:

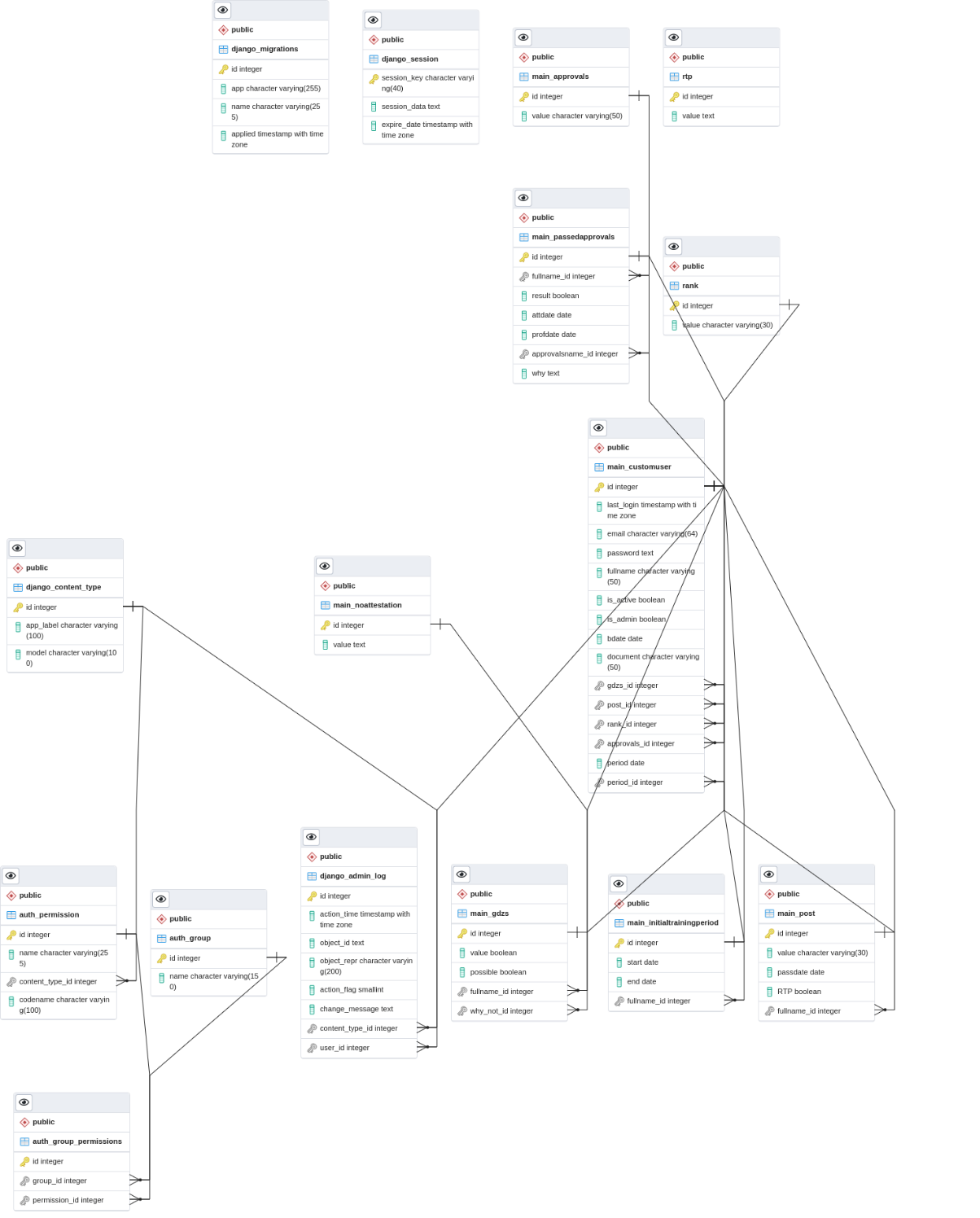


Рисунок 8 - представление базы данных проекта

Также в базе данных присутствуют таблицы, созданные самим Django. Они содержат данные и логи авторизации. Для правильного сохранения и обновления данных были написаны функции для БД, которые прилагаются к работе.

Ссылка на репозиторий с кодом: <https://github.com/mozgovoy/mchs>

С помощью системы контроля версий GIT отслеживались изменения проекта, а деплой на сервер производился с использованием GitHub. Хранение кода в веб-сервисе обезопашивает его от утери и обеспечивает доступ к нему с любого устройства в любое время.

## 2.4. UML-диаграммы

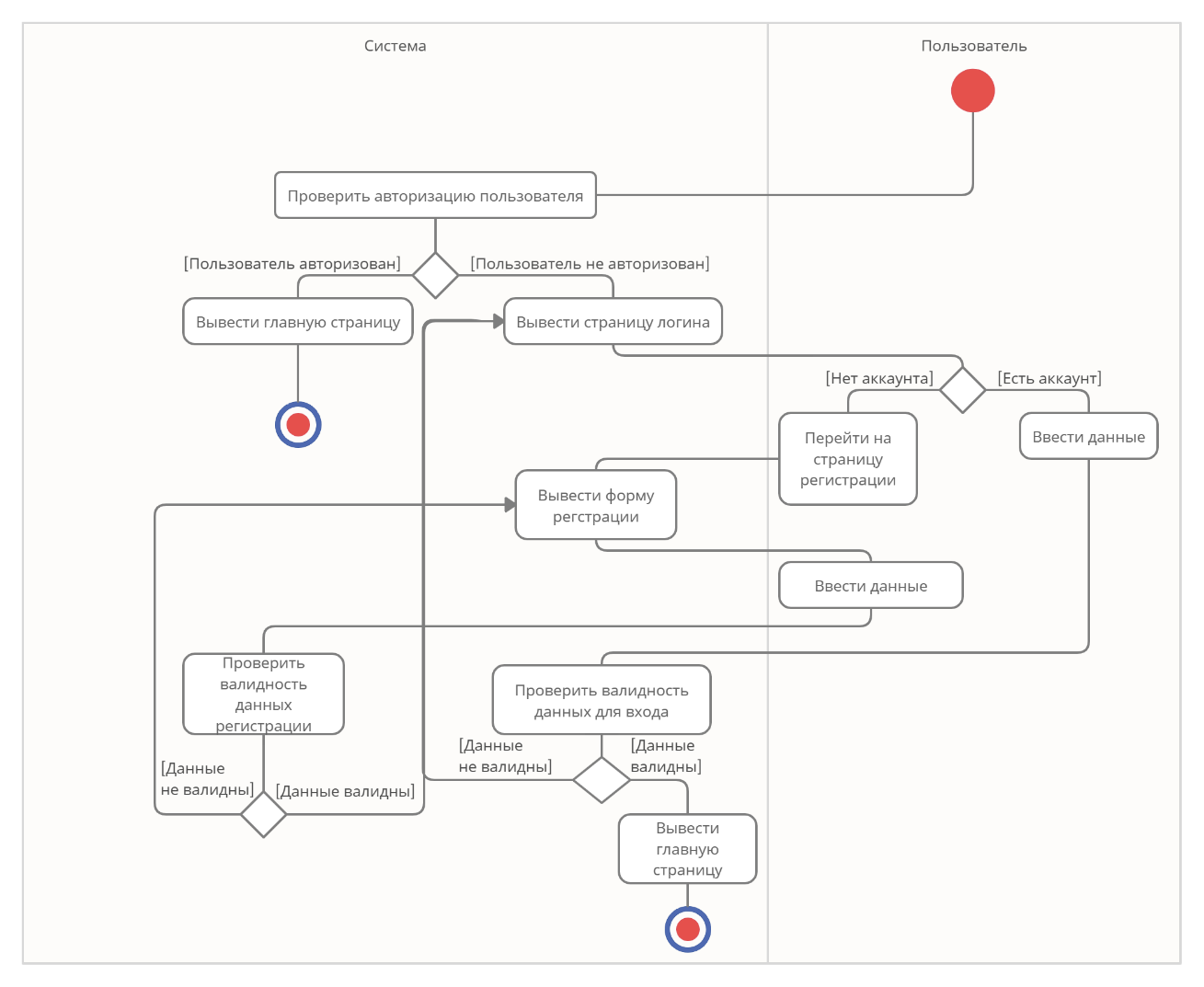


Рисунок 9 - диаграмма деятельности при авторизации пользователя

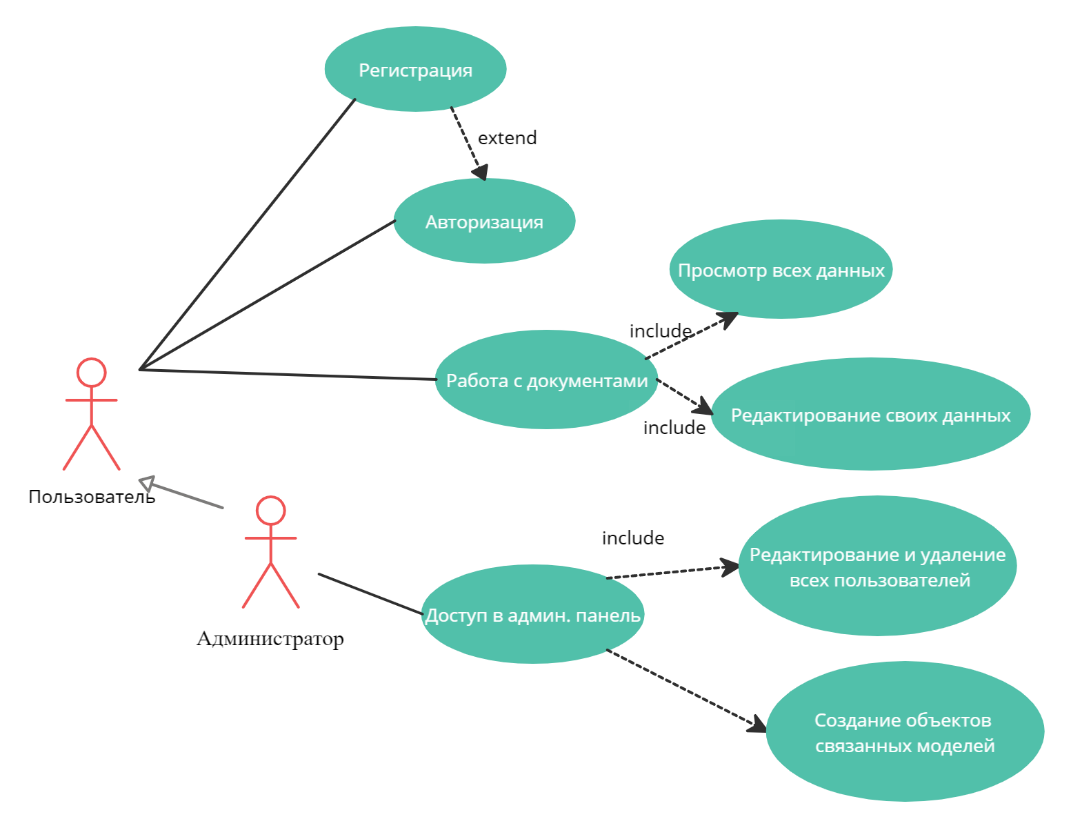


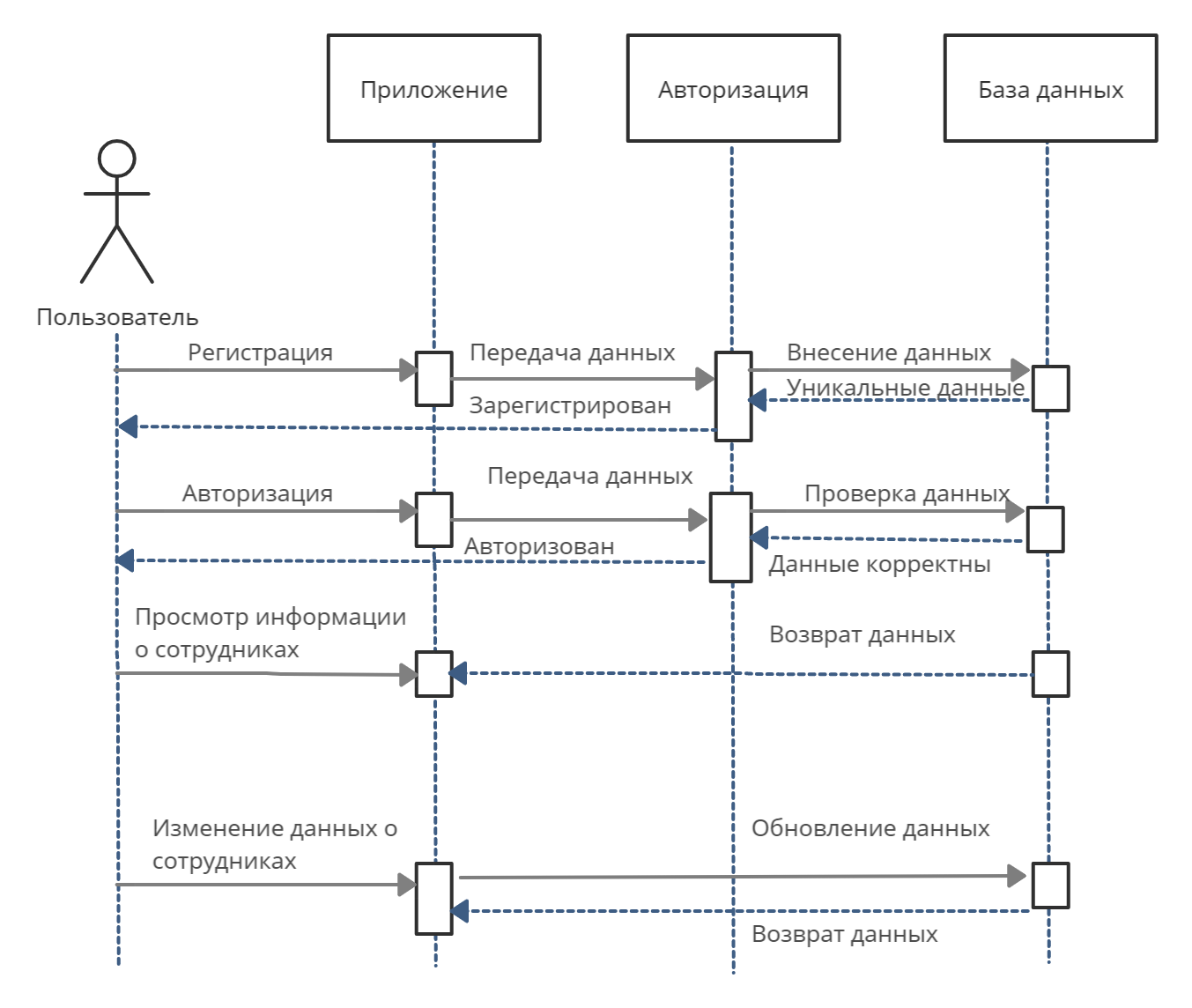
Рисунок 10 – диаграмма прецедентов 

Рисунок 11 – диаграмма последовательностей

# ЗАКЛЮЧЕНИЕ

В ходе выполнения работы была изучена информация по темам «Система электронного документооборота» и «Управление программным проектом». В процессе выполнения проекта были на практике изучены гибкая методология SCRUM, UML-диаграммы и управление программным проектом в целом. Разработка данного приложения с точки зрения программного продукта с соблюдением норм Agile дала полезный опыт для будущих проектов и работы в команде.

# ЛИТЕРАТУРА

1. Афанасьев, А. А. Веденьев, Л. Т. Воронцов, А.А. Аутентификация. Теория и практика обеспечения безопасного доступа к информационным ресурсам. - Москва: Горячая линия — Телеком, 2009. — 153 с.
2. Иванов, К. К. Проектирование базы данных. Роль процесса в создании информационной системы [Текст] / К. К. Иванов, А. А. Ефремов, И. А. Ващенко // Молодой ученый. — 2016. — № 18 (122). — С. 40-42.
3. Латыпова, Р. Т. Проблема соотношения бумажного и электронного документооборота [Текст] / Р. Т. Латыпова // Молодой ученый. — 2016. — № 1 (105). — С. 405-408.
4. Павлов, А. Е. Электронный документооборот в организации [Текст] / А. Е. Павлов // Молодой ученый. — 2020. — № 12 (302). — С. 126-128.
5. 1С:Документооборот. Документация [Электронный ресурс]. – Режим доступа: https://1c-rating.kz/programs/1c/doc/. – Дата доступа: 24.12.2021.
6. Kanban vs Scrum [Электронный ресурс]. – Режим доступа: https://www.atlassian.com/ru/agile/kanban/kanban-vs-scrum. – Дата доступа: 16.12.2021.
7. Wikipedia. Процесс разработки программного обеспечения [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Процесс\_разработки\_программного\_обеспечения. – Дата доступа: 19.12.2021.
8. Wikipedia. Процесс разработки программного обеспечения [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Процесс\_разработки\_программного\_обеспечения. – Дата доступа: 21.12.2021.
9. Wikipedia. Управление разработкой программного обеспечения [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Управление\_разработкой\_программного\_обеспечения. – Дата доступа: 21.12.2021.
10. КОНТУР.ДИАДОК [Электронный ресурс]. – Режим доступа: https://www.a-practic.ru/services/diadoc/. – Дата доступа: 26.12.2021.
11. Обзор системы электронного документооборота «Е1 Евфрат» [Электронный ресурс]. – Режим доступа: https://evfrat.ru/sed-e1-evfrat/obzor-sistemy/. – Дата доступа: 13.12.2021.

# ПРИЛОЖЕНИЕ А

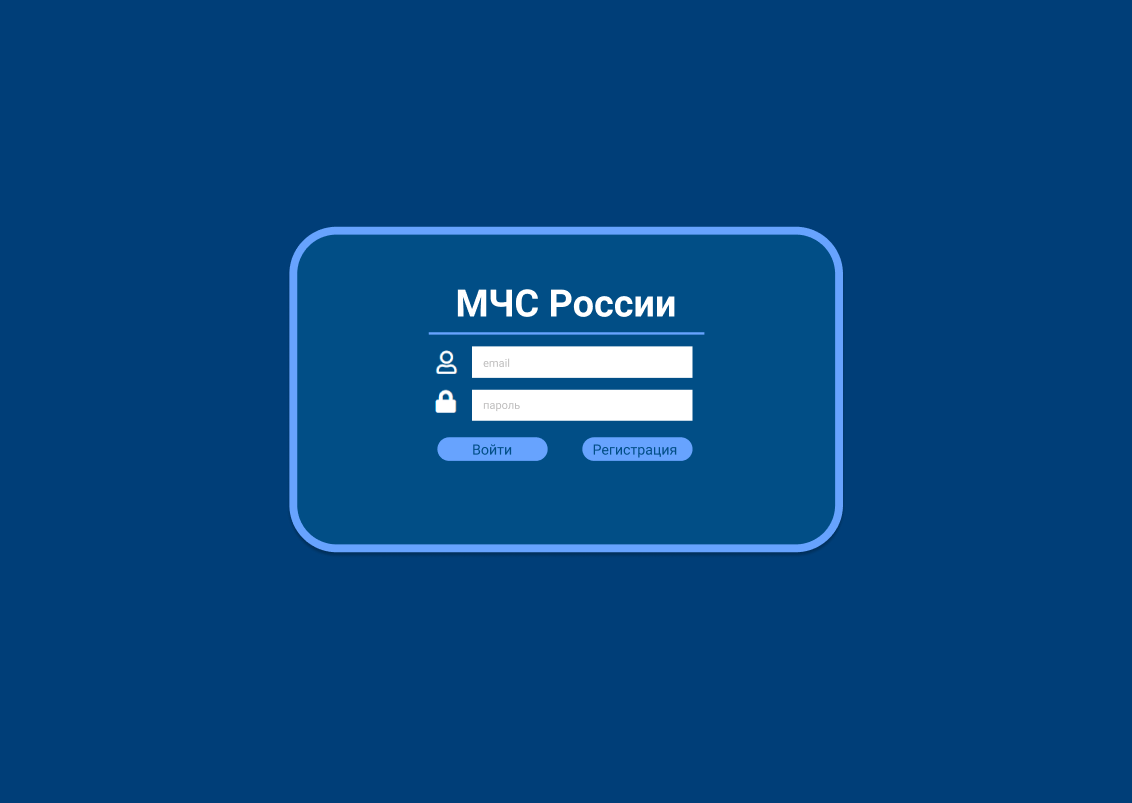


Рисунок 12 – макет страницы входа в учетную запись

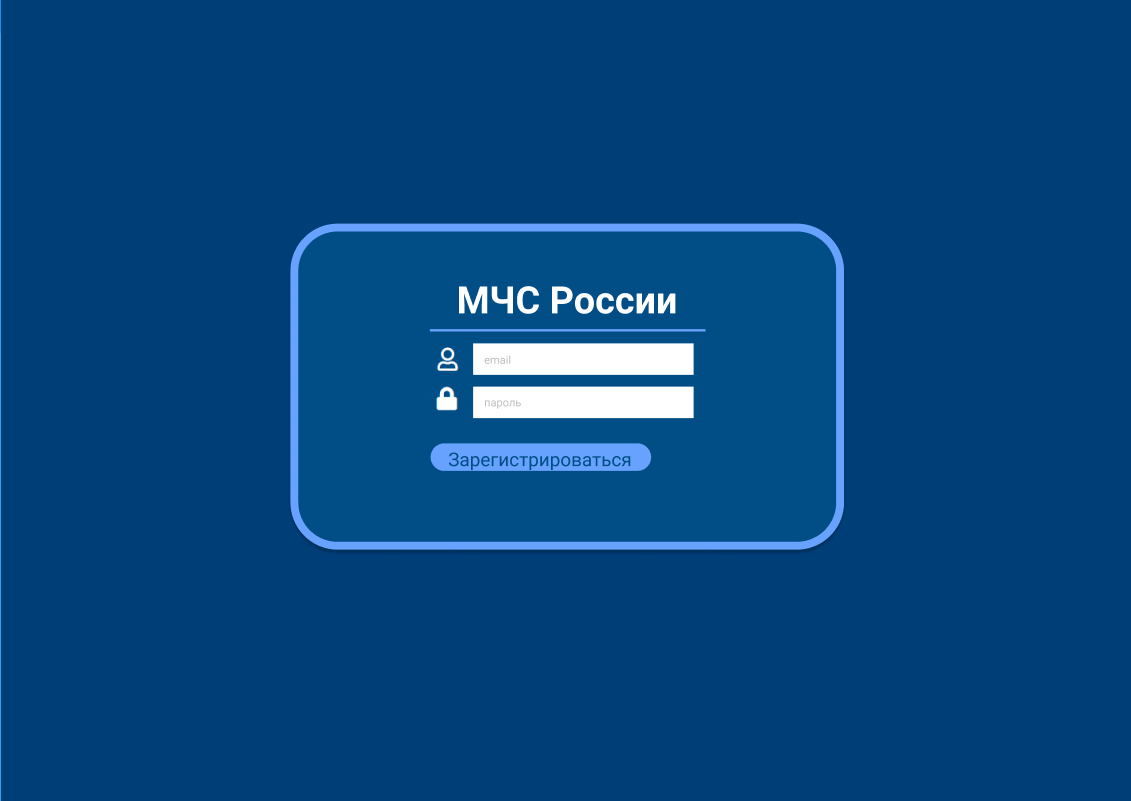


Рисунок 13 – макет страницы регистрации пользователя

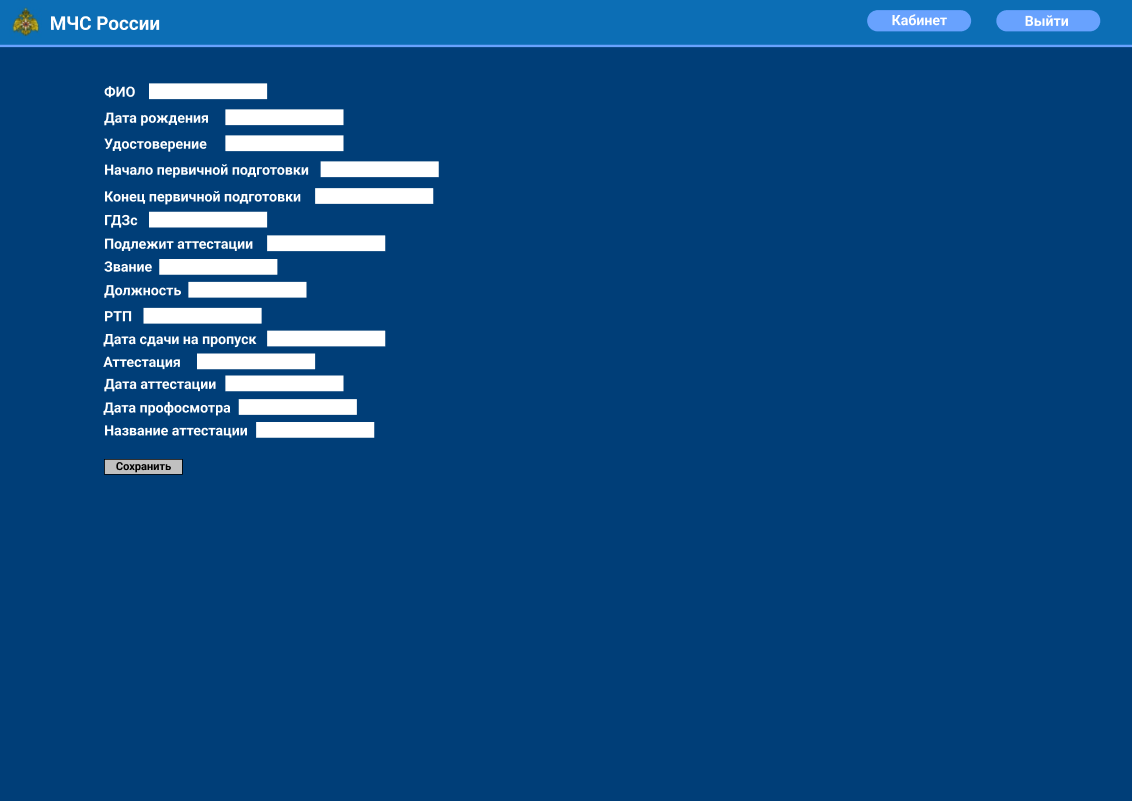


Рисунок 14 – макет страницы редактирования профиля

# ПРИЛОЖЕНИЕ Б

Триггер-скрипт для таблицы main\_approvals:

BEGIN

UPDATE main\_customuser SET

approvals\_id = new.id

WHERE id = new.fullname\_id;

RETURN new;

END;

Триггер-скрипт для таблицы main\_gdzs:

BEGIN

UPDATE main\_customuser SET

gdzs\_id = new.id

WHERE id = new.fullname\_id;

RETURN new;

END;

Триггер-скрипт для таблицы main\_period:

BEGIN

UPDATE main\_customuser SET

period\_id = new.id

WHERE id = new.fullname\_id;

RETURN new;

END;

Триггер-скрипт для таблицы main\_post:

BEGIN

UPDATE main\_customuser SET

post\_id = new.id

WHERE id = new.fullname\_id;

RETURN new;

END;