

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	9
1 АНАЛИЗ ЭЛЕКТРОННЫХ ПОДПИСЕЙ И СИСТЕМ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА.....	10
1.1 Электронная цифровая подпись .....	10
1.1.1 Существующие виды алгоритмов формирования ЭП .....	11
1.2 Коллективная электронная подпись .....	12
1.3 Система электронного документооборота .....	13
1.3.1 Влияние пандемии на рост рынка СЭД.....	15
1.4 Выводы.....	15
2 РАЗРАБОТКА АЛГОРИТМА КОЛЛЕКТИВНОЙ ЦИФРОВОЙ ПОДПИСИ.....	16
2.1 Анализ существующих алгоритмов.....	16
2.1.1 Стандарт ЭЦП – ГОСТ Р 34.10–94 .....	16
2.1.2 Стандарт ЭЦП – ГОСТ Р 34.10–2001 .....	17
2.1.3 Реализация множественной подписи на основе RSA .....	19
2.1.4 BLS подпись .....	19
2.2 Предложенный алгоритм .....	21
2.3 Выводы.....	26
3 РАЗРАБОТКА ЭКСПЕРИМЕНТАЛЬНОГО ОБРАЗЦА СИСТЕМЫ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА.....	27
3.1 Обзор аналогичных решений.....	27
3.2 Интеграция разработанного алгоритма подписи в экспериментальный образец .....	30
3.3 Выводы.....	32
4 АНАЛИЗ ЭФФЕКТИВНОСТИ РАЗРАБОТАННОЙ СИСТЕМЫ....	33

4.1 Выводы.....	35
ЗАКЛЮЧЕНИЕ .....	36
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	37
ПРИЛОЖЕНИЕ А Техническое описание .....	38

## ВВЕДЕНИЕ

В настоящее время протоколы электронной цифровой подписи повсеместно используются в системах информационных технологий, связанных с обменом и обработкой электронных сообщений и документов, имеющих различные юридические значения. Однако, существуют такие ситуации когда на документе требуются подписи нескольких пользователей системы, и в следствии этого возникают следующие проблемы:

- уменьшение скорости подписания документа;
- увеличение размера итоговой подписи;
- долгая проверка подписи;
- возможность идентифицировать других участников подписи.

В организациях, использующих системы электронного документооборота, сложилась многолетняя традиция оставлять подписи на углах экземпляров документов. Только после того, как все ответственные лица, имеющие отношение к необходимому документу, поставят свои подписи, документ продвигался дальше по иерархии документооборота и попадал на стол для утверждения директору.

Существующие алгоритмы не позволяют полностью повторить данную процедуру с обеспечением важного свойства, а именно, чтобы сторонние лица не могли определить сотрудников, которые визировали документ, но в случае проверки руководитель мог доказать, ответственных людей, которые подписывали данный документ, стороннему проверяющему.

Для решения такой задачи целесообразно создать систему, предоставляющую функции электронного документооборота, с возможностью использования коллективной цифровой подписи.

# 1 АНАЛИЗ ЭЛЕКТРОННЫХ ПОДПИСЕЙ И СИСТЕМ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА

## 1.1 Электронная цифровая подпись

В текущее время во многих организациях предпринимаются попытки, либо уже используются различные способы безбумажного оборота или обмена документами. Данные системы позволяют значительно сократить время, используемое для обмена и обработки документации, усовершенствовать и произвести уменьшение затрат на процедуру подготовки, пересылки, учета и хранения документов, создать систему для обмена документами внутри организации. Однако при изменении привычного документооборота на электронный, появляется необходимость в знании автора документа, достоверности данных в документе и защиты от различного рода искажений находящейся информации в документе. Эта проблема решается электронной цифровой подписью, представляющее из себя средство защиты документа от искажений и позволяющая однозначно идентифицировать подписанта документа.

Электронная цифровая подпись – это часть электронного документа [1], которая получается в результате криптографического преобразования информации и позволяющая проверить следующие факторы:

- целостность – есть отсутствие искажений информации в документе с момента формирования подписи,
- авторство – принадлежность подписи владельцу сертификата ключа подписи;
- неотказуемость – в случае успешной проверки подтвердить факт подписания электронного документа.

Применение электронной подписи позволяет сократить затрачиваемое время на обмен и форматирование документации, а так же улучшить процесс хранения и учета, в частности, ускорить поиск необходимых документов.

По своим функциям цифровая подпись имеет много общего с обычной рукописной подписью и обладает ее основными достоинствами:

- предоставляет гарантии целостности подписанного текста;
- удостоверяет, что подписанный текст появился от лица, поставившего подпись;
- не дает подписанту возможности отказаться от обязательств, связанных с подписанным текстом.

### 1.1.1 Существующие виды алгоритмов формирования электронной цифровой подписи

На данный момент существует два основных варианта схем построения алгоритмов электронной цифровой подписи:

- а) Алгоритмы, на основе схем симметричного шифрования [2].

Данные схемы требуют при процессе формирования подписи присутствие третьего лица, которому участники доверяют процедуру подписи. Далее происходит шифрование документа секретным ключом и передача третьему лицу. В основе симметричных алгоритмов лежат блочные шифры.

Симметричные схемы имеют следующие преимущества:

- 1) На данный момент, хорошая изученность блочных шифров, является следствием и гарантом стойкости используемых симметричных схем.
- 2) Если существует необходимость увеличения стойкости алгоритма, возможно с минимальными изменениями самой реализации, довольно легко заменить шифр на более стойкий.

Недостатки систем с использованием симметричных шифров:

- 1) Существенное увеличение подписи из-за необходимости отдельного подписывания каждого бита информации.
- 2) Подпись может иметь больший размер, чем само сообщение на несколько порядков.

3) Ключи, сгенерированные для подписи, из-за раскрытия половины секретного слова после совершения подписи, могут быть использованы только один раз.

б) Алгоритмы, в основе которых лежат схемы асимметричного шифрования [3].

Один из пользователей системы хочет подписать цифровой документ и переслать его другому участнику документооборота. Есть третье лицо, которое является посредником, владеющим ключами для секретной связи с пользователями. Данные ключи устанавливаются до начала протокола обмена и используются множество раз. Симметричный шифр используется для шифрования передаваемого сообщения.

Особенности протокола:

1) Только один пользователь и третье лицо имеют секретный ключ для общения между ними, поэтому только первый пользователь может отправить посреднику сообщение, зашифрованное на первом ключе. (защита ЦП от подделки).

2) Невозможно изменить уже подписанный документ, с сохранением текущей подписи. Для каждого нового документа необходимо производить новую подпись.

3) Реализуется неотказуемость от документа в следствии подписания.

## 1.2 Коллективная электронная подпись

Индивидуальная подпись является наиболее широко применяемым видом электронной цифровой подписи. Однако, существуют такие ситуации, когда для документа нужно иметь больше, чем одну подпись. В частности, задачи передачи или верифицирования документов от совокупности людей или от имени коллегиального органа, делают актуальной идею о разработке коллективной электронной цифровой подписи [4].

На текущий момент, системы электронного документооборота не позволяют подписывать электронный документ более чем одному

пользователю [5], что уменьшает время эффективного использования, обработки и подписания документа, при использовании документа большим количеством пользователей системы. Из этого следует, что размерность подписи становится в несколько раз больше, пропорционально количеству участников системы, подписывающих этот электронный документ, при использовании обычных видов подписей. Более того, для проведения процедуры проверки валидности электронной подписи, необходимо произвести проверку данных всех участников процесса.

Реализация метода создания и верификации коллективной электронной цифровой подписи даст возможность нескольким пользователям системы производить обработку и подписание документа. При этом длина и количество подписей не увеличивается, что дает возможность уменьшить объем избыточной информации, необходимой для верификации электронных документов, а также увеличить защищенность подписываемого документа.

### 1.3 Система электронного документооборота

В последнее десятилетие отечественная отрасль систем электронного документооборота совершила колоссальный путь. Системы для совершения обычного делопроизводства в небольших компаниях, эволюционировали до систем класса ЕСМ, используемые в огромных корпорациях, составляя конкуренцию всем остальным участникам рынка.

В 2019 году Российский рынок систем электронного документооборота и ЕСМ-систем продолжил планомерное развитие, без необычного большого увеличения количества новых клиентов. Эксперты в этой области отмечают, что в последнее время, компании занимаются качественной модификацией и плавной трансформацией уже существующих решений, а также о создании отдельных специализированных программ, вместо создания новых продуктов для массового внедрения в новые организации.

По оценке TAdviser [6], объем отечественного рынка СЭД/ЕСМ-систем по итогам 2019 года увеличился на 8% и достиг отметки в 52,4 млрд рублей, что представлено на рисунке 1.



Рисунок 1 – объем российского рынка СЭД

Большинство опрошенных участников отрасли говорят что 2020 год в большей своей части прошел с положительным результатом для всей сферы СЭД/ЕСМ-систем. Многие Ожидают, что положительная динамика не только сохраниться, но и прибавит в своем эффективном росте, по отношению к 2019 году. Специалисты отмечают, что установка СЭД на предприятии не стала задачей номер один, в отличии, например, от сферы видеоконференцсвязи, но тем не менее они оказались менее затронуты кризисом, чем другие продукты.

В следствии пандемии 2020 года, произошел массовый переход на удаленную работу. Это событие позволило продемонстрировать всю важность использования электронного документооборота для обычного функционирования бизнеса [7], находящегося в стрессовой ситуации. В следствии этого ожидается дальнейшая положительная динамика роста рынка этой отрасли.

Также, ещё одним мощным, но не единственным, фактором, способным увеличить положительный рост динамики развития систем электронного документооборота в нашей стране является импортозамещение. Но теперь оно проявляется не в острой необходимости появления хоть каких-нибудь



российский решений, а в стабильной необходимости в оптимальных решениях со стороны крупных организаций. Отечественные решения становятся всё сильнее, по сравнению с зарубежными аналогами, а новые системы, более адаптированными под российскую специфику рынка, а также самого документооборота.

### 1.3.1 Влияние пандемии на рост рынка СЭД

Из-за пандемии 2020 года, стал очевиден результат оценки важности цифровизации бизнес-процессов [6], и насколько важно обеспечивать непрерывность работы этих процессов в условиях непредвиденных чрезвычайных ситуаций. Подходящая система электронного документооборота показала себя в качестве превосходного инструмента для решения повседневных задач, различных направлений оборота, выходящих за рамки канцелярии, регистрации и согласования документов.

Без использования развитой системы документооборота невозможно отказаться от обмена бумажными документами внутри компании, а также поддержки работы сотрудников не находясь в офисе компании, что очень необходимо компаниям, находящимся в условиях карантина и пандемии.

Ещё одним следствием кризиса пандемии, является то, что в 2020 году рынок рассматриваемой сферы получил второе дыхание. Компании, которые обдумывали возможность применения такой системы, либо начали использовать её, либо ушли с рынка. Заказчики более внимательно оценивают выгоду от использования продукта, и используют преимущества автоматизации в тех областях бизнеса, в которых это принесёт наибольшую выгоду и эффективность.

## 1.4 Выводы

В данной главе рассмотрены функции электронной цифровой подписи, существующие виды алгоритмов, приведено описание коллективной цифровой подписи, а также проведён анализ рынка СЭД.

## 2 РАЗРАБОТКА АЛГОРИТМА КОЛЛЕКТИВНОЙ ЦИФРОВОЙ ПОДПИСИ

### 2.1 Анализ существующих алгоритмов

Для понимания структуры разрабатываемого алгоритма есть необходимость в рассмотрении уже существующих реализаций.

#### 2.1.1 Стандарт ЭЦП – ГОСТ Р 34.10–94

Данный стандарт [8] регламентирует использование числа  $p$ , которое должно быть простым, двоичным и находится в границах  $510 \leq |p| \leq 512$  бит, или  $1022 \leq |p| \leq 1024$  бит. Число  $(p - 1)$  содержит просто большой делитель  $q$ , такой что  $2^{255} \leq q$  и  $q \leq 2^{256}$  для первой границы, или же для второй границы данное выражение равно  $2^{511} \leq q$  и  $q \leq 2^{512}$ . Для верификации используется число  $\alpha \neq 1$ , такое что  $\alpha^q \bmod p = 1$ , где  $\alpha$  – генератор подгруппы достаточно большого простого порядка  $q$ .

Для высчитывания ЭЦП используется алгоритм:

1. Создается случайное число  $k$ ,  $1 < k < q$ .
2. Высчитывается значение  $R = (\alpha^k \bmod p) \bmod q$ , представляющее из себя первую часть подписи.
3. От подписываемого сообщения высчитывается хэш-значение, по ГОСТ Р 34.11–94.
4. Высчитывается

$$S = kH + zR \bmod q,$$

где  $z$  – секретный ключ. При  $S = 0$ , создание подписи повторяется. Данное значение является второй частью подписи.

Для верификации подлинности нужно проделать следующие шаги:

1. Подпись не является действительной если не выполняются условия  $r < q$  и  $s < q$
2. Вычисляется значение

$$R' = (\alpha^{S/H} y^{R/H} \bmod p) \bmod q, \quad (1)$$

где  $y$  – является открытым ключом пользователя.

3. Значения  $R$  и  $R'$  проверяются на идентичность. Если они эквивалентны, то подпись считается верной.

Для создания коллективной электронной цифровой подписи, каждый  $i$ -й пользователь системы формирует свой открытый ключ, в виде

$$y^i = \alpha^{z_i} \bmod p,$$

где  $z_i$  – это личный секретный ключ,  $i = 1, 2, \dots, m$ .

Произведение  $y = y_1 y_2 \dots y_m \bmod p$  является коллективным открытым ключом.

Для формирования коллективной подписи каждый участник процесса подписи выбирает разовый случайный секретный ключ  $k_i$ , затем вычисляет  $R_i = (\alpha^{k_i} \bmod p) \bmod q$ , предоставляя это значение для коллективного использования. После этого вычисляется произведение  $R = R_1 R_2 \dots R_m \bmod q$ , которое используется каждым пользователем для нахождения своей части подписи, по формуле  $S_i = k_i H + z_i R \bmod q$ .

Пара чисел  $(R, S)$  является коллективной подписью, где  $S$  находится по формуле  $S = S_1 + S_2 + \dots + S_m \bmod q$ .

Для проверки используется формула (1), Если  $R = R'$ , то подпись от группы  $m$  пользователей системы является подлинной, так для создания подписи нужен секрет каждого участника процесса создания подписи.

### 2.1.2 Стандарт ЭЦП – ГОСТ Р 34.10–2001

Данный стандарт [9] регламентирует использование числа  $p$ , являющимся модулем эллиптической кривой, которая задается в систем координат уравнением

$$y^2 = x^3 + ax + b \bmod p,$$

где  $a$  и  $b \in GF_p$  ( $GF_p$  – это поле Галуа порядка  $p$ ).

Точка  $G$  не должна совпадать с началом координат, а произведение  $qG$  – совпадает. Число  $d$ , которое представляет из себя большое целое число, является секретным ключом, а открытым – точка  $Q = dG$ . Подпись  $(R, S)$  создается по алгоритму:

1. Выбирается такое случайно число  $k$ , что  $0 < k < q$ .
2. Вычисляются координаты точки  $C = kP$  на эллиптической кривой и находится значение

$$R = xC \bmod q,$$

где  $x_C$  – координата точки  $C$ .

3. Вычисляется значение  $S = (Rd + ke) \bmod q$ , где  $e = H \bmod q$ .

Пара чисел  $(R, S)$  являются подписью.

Для выяснения валидности подписи, необходимо вычислить координаты точки эллиптической кривой:

$$C = ((Se^{-1}) \bmod q)G + ((q - R)e^{-1} \bmod q)Q, \quad (2)$$

а также определение значения  $R' = x_C \bmod q$  и проверку выполнения равенства  $R' = R$ .

Для формирования коллективной подписи каждый участник  $i$ -й процесса подписи формирует открытый ключ вида  $Q = d_i G$ , где  $d_i$  – личный (секретный) ключ,  $i = 1, 2, \dots, m$ . Открытым коллективным ключом будет сумма  $Q = Q_1 + Q_2 + \dots + Q_m$ .

Для формирования коллективной подписи каждый участник процесса подписи выбирает разовый случайный секретный ключ  $k_i$ , затем вычисляется  $C_i = k_i G$  и передает их для коллективного использования. Далее вычисляется сумма всех точек  $C = C_1 + C_2 + \dots + C_m$ , из которой вычисляется значение  $R$ . Каждый пользователь вычисляет свою часть подписи  $S_i = (Rd_i + k_i e) \bmod q$ .

Пара чисел  $(R, S)$  является коллективной подписью, где  $S$  вычисляется по формуле  $S = S_1 + S_2 + \dots + S_m \bmod q$ .

Для проверки используется формула (2), Если  $R = R'$ , то подпись от группы  $m$  пользователей системы является подлинной, так для создания подписи нужен секрет каждого участника процесса создания подписи.

### 2.1.3 Реализация множественной подписи на основе RSA

Схема двойной цифровой [10] подписи расширяет обычную схему RSA. Вместо того, чтобы создавать пару ключей (открытый и закрытый ключ), создается тройка ключей (один публичный и два приватных). Аналогично с схемой RSA, участники выбирают модуль вычисления числа  $n$  – произведение двух простых длинных чисел. Выбираются два случайных приватных ключа  $r$  и  $s$ , в диапазоне от 1 до  $n$ , которые будут взаимно простые с  $\varphi(n)$ , где  $\varphi(n)$  – функция Эйлера.

Открытый ключ вычисляется по формуле  $r * s * t = 1 \bmod \varphi(n)$ . Число  $t$  будет являться публичным ключом. Для того чтобы подписать величину  $C$ , первый участник вычисляет  $S_1 = C^r \bmod n$ . Результат этого вычисления передается второму участнику, у которого появляется возможность увидеть подписываемое сообщение. Он получает величину  $C$  из величины  $S_1$ . Для подписания необходимо вычислить  $S_2 = S_1^s \bmod n$ . Подпись проверяется с помощью  $C = S_2^t \bmod n$ .

Для расширения схемы двойной подписи до  $n$  участников, создаются  $n$  случайных приватных ключей  $k_1, k_2, \dots, k_n$ . Для вычисления публичного ключа, используется формула  $(k_1 + k_2 + \dots + k_n) * t = 1 \bmod \varphi(n)$ . Каждый  $i$ -й участник подписывает сообщение  $M$  по формуле  $S_i = M^{k_i} \bmod n$ . Затем вычисляется величина  $S = S_1 * S_2 * \dots * S_n \bmod n$ .

Подпись проверяется по формуле  $S^t \bmod n = (S_1 * S_2 * \dots * S_n)^t \bmod n = M^{(k_1 + k_2 + \dots + k_n) * t} \bmod n = M$ .

### 2.1.4 BLS подпись

Из-за работы подписи с эллиптическими кривыми [11], возникает необходимость использования особенной функции хеширования, такой чтобы

после применения функции мы имели координаты точки на кривой. В качестве основной функции используется стандартная функция хэширования, в результате которой мы получаем обычное хэш-значение, которое будем считать координатой  $x$  кривой. У каждого значения  $x$  может быть соответствующее значение ноль или два значения  $y$ , то есть не для каждого  $x$  есть валидное значение  $y$ . По этой причине сообщение хэшируется  $(msg \parallel i)$ , пока оно не выдаст корректный результат, где  $\parallel$  – конкатенирующая функция, а  $i$  – положительное число. В конце выбирается одна из двух полученных точек, например большее из значений  $y$ .

Для возможности произвести подпись, используется функция сопоставления двух точек на кривой к некоторому числу. Вводится определение спаривания. Пусть  $G, G_t$  – циклические группы простого порядка  $r$ , порожденные элементом  $g$ . Функция называется эффективно вычислимой  $e : G_1 \times G_2 \rightarrow G_T$ , для которой выполняются следующие свойства:

1. Невырожденность:  $e(g, g) \neq 1$
2. Билинейность:  $e(g^a, g^b) = e(g, g)^{ab}$ , где  $a, b \in \mathbb{Z}$

Для использования в криптографии распространенными являются функции спаривания Тейта, Вейля и оптимальное спаривание Эйта.

Когда для циклической группы определена функция спаривания, то для этой группы неразрешимы вычислительная задача Диффи-Хеллмана и задача дискретного логарифма, однако существует эффективное решение задачи принятия решения Диффи-Хеллмана. Такие группы называют группами Диффи-Хеллмана и подразумевают схему подписи, называемую подписью Боне – Линна – Шахама (BLS).

Пусть  $G$  – это группа Диффи-Хеллмана простого порядка  $r$ , где  $g \in G$  – порождающий элемент группы,  $m$  – заданное сообщение. Тогда, для создания закрытого ключа  $SK$ , выбирается случайное целое число, находящееся в интервале  $[0, r - 1]$ , а открытым ключом будет  $PK = g^{sk}$ .

Для создания подписи необходимо хэшировать сообщение в Кривую

$$H = \text{Hashing}(m),$$

где  $H$  – это точка на кривой. После этого вычисляется  $S = h^{SK}$ , которая и является подписью сообщения.

Для проверки подписи вычисляется  $d_1 = e(PK, H)$ , после этого находится значение  $d_2 = e(g, S) = e(g, H^{SK}) = e(g^{SK}, H)$ . Далее сравниваются значения  $d_1$  и  $d_2$ , и если они совпадают, то подпись считается верифицированной.

Для того чтобы произвести агрегирование ключей, предположим, что мы имеем группу подписей, которая имеет  $n$  пар  $(S_i, PK_i)$ , где  $i = [0, n]$ . Коллективной подписью системы назовём сумму  $S_i$  по  $i$ . Для проверки подписи необходимо подтвердить равенство  $e(g, S) = e(PK_1 H_1) \times e(PK_2 H_2) \times \dots \times e(PK_n H_n)$ .

Для верификации нет необходимости знать сообщения, соответствующие индивидуальным подписям, однако необходимо знать публичные ключи и выполнить операцию спаривания  $n + 1$  раз.

## 2.2 Предложенный алгоритм

Исходя из рассмотренных аналогов было решено создать алгоритм на основе схемы BLS. Эта схема позволяет объединять множество подписей в одну, без дополнительных коммуникационных циклов. Нет нужды полагаться на генераторы случайных чисел в самом алгоритме. Сама же схема является очень простой и понятной.

Один из основных недостатков данного типа подписи является процесс спаривания. Этот процесс является проблемой в блокчейн схемах, так как вычисление спаривания занимает некоторое время, однако в нашем случае это не является минусом, так как использование этой схемы преследует другие цели.

Схематичное представление структуры предлагаемого алгоритма представлено на рисунке 2.

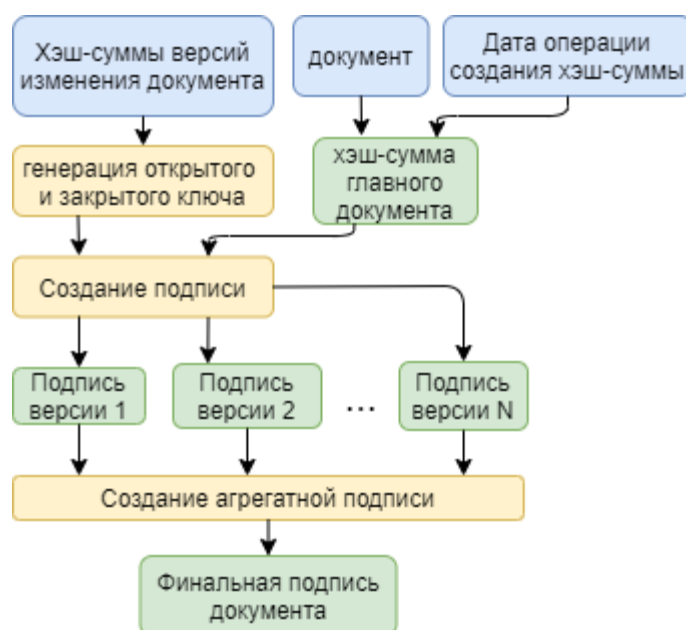


Рисунок 2 – Схема предлагаемого алгоритма

Каждая версия документа, сохраненная на веб ресурсе, имеет свой публичный и секретный ключ, который является хэш-суммой версии документа. Для создания подписи документа создается хэш-сумма документа, совмещённая с датой начала операции по созданию подписи. Для предыдущих версий так же создаются подписи. Данная хэш сумма будет являться подписываемым сообщением. После этого, используя открытый ключ каждый предыдущей версии документа, создается подпись. Агрегатная подпись всех созданных подписей и будет являться финальной подписью.

Для создания подписи с несколькими участниками используется такая же схема, только для каждого пользователя создается индивидуальная подпись, которая потом вшивается в общую. Схематичное представление такого алгоритма представлено на рисунке 3.



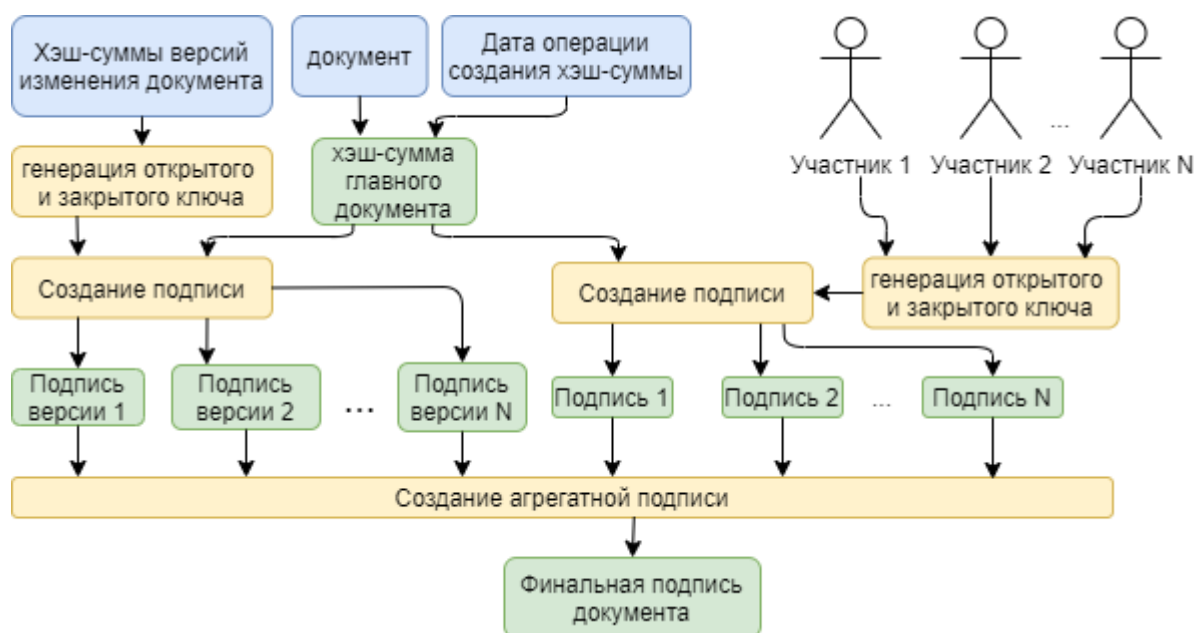


Рисунок 3 – Схема подписи с дополнительными участниками

Схематичное представление функции алгоритма подписи представлено на рисунке 4.

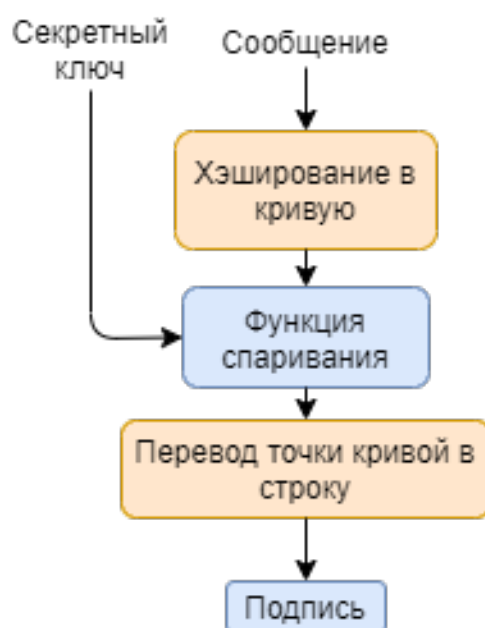


Рисунок 4 – Схема подписи

Сообщение, которым является хэш-сумма документа, подается на вход подписывающей функции. Производится перевод этого значения в точку на кривой, которая, с помощью функции спаривания, соединяется с секретным

ключом, который переводится из точки в строку, которая и будет являться подписью документа.

Производится перевод этого значения в точку на кривой, соединяющуюся с секретным ключом с помощью функции спаривания. Ключ переводится из точки в строку, являющейся подписью документа.

Схематичное представление алгоритма верификации представлено на рисунке 5.

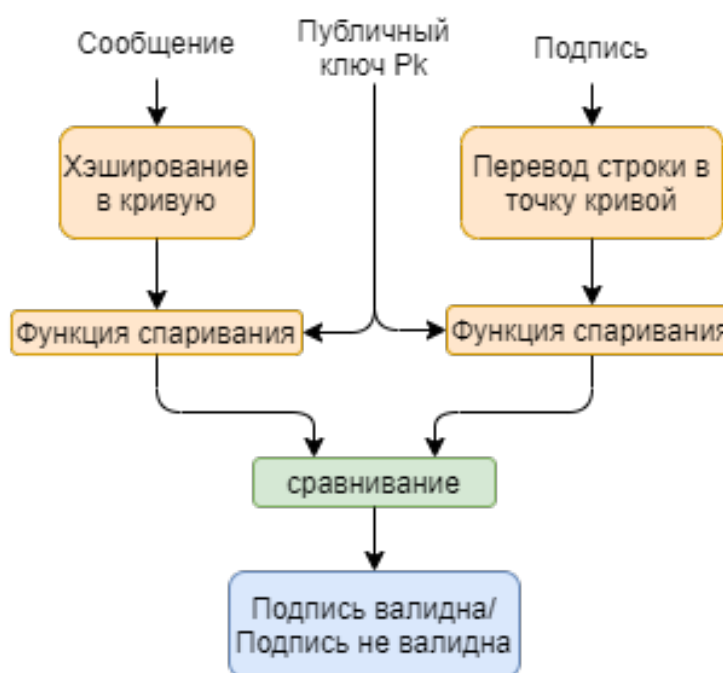


Рисунок 5 – Схема проверки подписи

Для подтверждения валидности подписи, хэш-сумма документа переводится в точку на кривой. Тоже самое происходит и с проверяемой подписью. Обе эти точки подаются в свои функции спаривания, вместе с публичным ключом, который предоставляется для проверки подписи. После этого происходит сравнения полученных значений, и на основе этого делается результат о валидности подписи.

Схематичное представление алгоритма агрегации ключей представлено на рисунке 6.

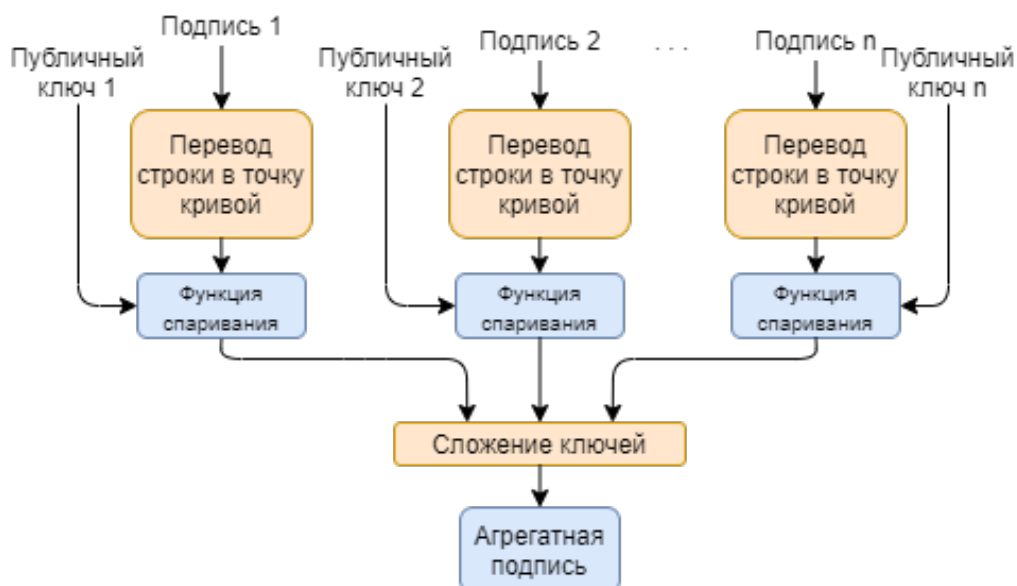


Рисунок 6 – Схема агрегации ключей

Для создания агрегатной подписи используются похожие этапы, однако, из-за особенности использования эллиптических кривых, существует возможность складывать полученные значения. Финальная подпись получается путем сложения всех значений, полученных из функций спаривания.

Схематичное представление алгоритма проверки агрегированной подписи представлено на рисунке 7.

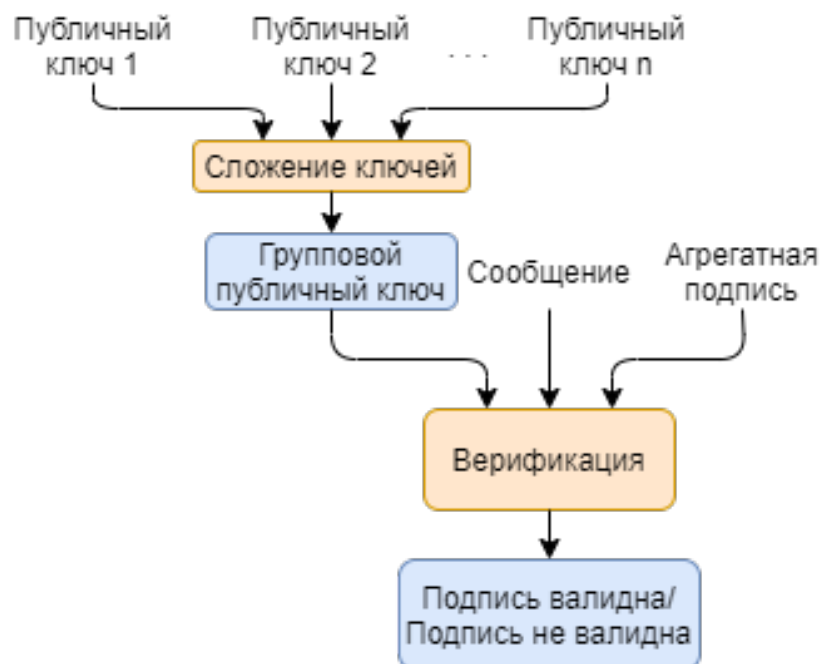


Рисунок 7 – Схема проверки групповой подписи

Для проверки агрегатной подписи, на вход обычной функции проверки подается агрегатная подпись, хэш-значение документа и групповой публичный ключ, который получается путем сложения всех публичных ключей, используемых для создания агрегатной подписи.

## 2.3 Выводы

В данной главе приведены анализы существующих алгоритмов. С помощью сравнения этих решений был предложен собственный алгоритм на основе схемы BLS.

### 3 РАЗРАБОТКА ЭКСПЕРИМЕНТАЛЬНОГО ОБРАЗЦА СИСТЕМЫ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА

#### 3.1 Обзор аналогичных решений

Системы электронного документооборота выполняют задачи по целостному хранению, обеспечению бесперебойного доступа, поиску среди всех документов, регистрации новых документов, контролю исполнения, согласования и так далее. У различных организаций имеются различные интересы и требования к системе, обработка документов в государственной организации значительно отличается от банковской или архивной структуры.

Для государственных организаций наиболее важным является возможность регистрации и контроля исполнения заданий и поручений по документам, а для архивного делопроизводства важными функциями являются поиск и построение структуры информации, для банков же – это обработка и пересылка данных клиентов и интеграция с автоматизированными банковскими системами. Функциональные особенности системы электронного документооборота сильно зависят от размера организации:

- в небольшой компании один сотрудник выполняет все операции с документами, регистрирует их, а также отмечает факт их исполнения;
- большая организации имеет много различных сотрудников, которые занимаются регистрацией, согласованием, исполнением и так далее.

Из-за существования таких требований, существует необходимость создания отдельных рабочих пространств и реализация возможности разграничения доступа к документу.

Фактически, одни и те же реализации функций невозможно использовать в небольшой и в крупной компании: в малой компании потребуются выполнять слишком большое количество действий, а в крупной невозможно обеспечить правильный доступ к документу. С учетом всего сказанного, можно говорить о том, что уникальных решений для автоматизации работы в любой

организации с документами не существует и нет возможности создать такое решение. Есть набор стандартных функций, выполняемых в системе электронного документооборота, но реализация каждой функции будет зависеть от конкретных задачи самой системы и её пользователей.

Исходя из выше сказанного, так как разработка не создается для какой-либо конкретной организации, было решено провести обзор среди существующих бесплатных аналогов, а также решений с временным доступом.

«Alfresco» – распространенная в западных странах система электронного документооборота и управления проектами. Отличительные особенности:

- 1) открытость кода (полный Open Source);
- 2) отличается от большинства продуктов с открытым кодом стабильностью работы;
- 3) возможности масштабирования под задачи бизнеса любого размера.

«FossLook» - это платформа для построения решений по управлению документооборотом и работе с ним. Это система состоит из решений:

- 1) электронного документооборота FossDoc;
- 2) управления взаимоотношениями с клиентами FossLook CRM;
- 3) систем управления заявками FossLook Заявки;
- 4) и другие.

Возможности проекта FossLook можно использовать, создавая для себя собственные решения или расширяя существующие. Специалисты компании оказывают помощь в подготовке решений на базе FossLook. Используя FossLook, решаются задачи формализации документов, хранения, целостности, совместной работы с документами, включая согласование, исполнение и ознакомление, использование электронной цифровой подписи.

«АВРОРА: Документооборот» - предназначена для автоматического обмена документами в государственных и коммерческих предприятиях различного рода деятельности. Автоматизируется весь жизненный цикл документов.

Система предоставляет инструменты работы с информационными потоками (документацией, информацией, файлами) вне зависимости от территориальной удаленности подразделений, создающий единое пространство информационного взаимодействия между сотрудниками.

Система масштабируется в зависимости от размеров и потребностей организации.

«Астрал Онлайн» - система электронного документооборота, которая соответствует современным требованиям безопасности. Обмен значимыми документами с контрагентами, организация документооборота внутри компании. «Астрал Онлайн» позволяет пользоваться системой с любого устройства с выходом в интернет и имеет роуминг со всеми ведущими системами электронного документооборота.

«NauDoc» - программный продукт для автоматизации делопроизводства, документооборота и бизнес-процессов, разработанный компанией NAUMEN. NauDoc позволяет управлять процессами обработки документов и осуществлять контроль исполнительской дисциплины, получая исчерпывающую информацию о ходе выполнения заданий. Раньше распространялась в свободном формате, но в настоящее время продукт сняли с производства.

В процессе обзора аналогичных решений выявлены следующие функциональные параметры, по которым будет произведено их сравнение в таблице 1:

- web-приложение;
- desktop-приложение;
- редактор документа;
- работа с почтой;
- роуминг с другими системами;
- цифровая подпись;
- коллективная цифровая подпись;

- использование сертификационных центров;

Таблица 1 - Сравнение аналогов.

	Alfresco	FossLook	«АВРОРА»	«Астрал Онлайн»	«NauDoc»
web-приложение	+	-	-	+	-
desktop-приложение	+	+	+	-	+
Редактор документа	+	-	+	-	-
Работа с почтой	+	+	В платной версии	-	+
Роуминг с другими системами	-	-	-	+	-
цифровая подпись	+	+	+	+	+
коллективная цифровая подпись	-	-	-	-	-
использование сертификационных центров	+	+	+	+	+

Важно отметить, что хоть рассмотренные системы и поддерживают электронные подписи, однако все реализации используют различные внешние системы и центры. Таким образом, создание системы без использования сертификационных центров является обоснованным.

Техническое описание разработанного продукта представлено в приложении 1.

### 3.2 Интеграция разработанного алгоритма подписи в экспериментальный образец

Предложенный алгоритм создания подписи используется как внешний модуль системы. После того, как пользователь вызовет функцию создания подписи с помощью интерфейса системы, все необходимые ресурсы для работы алгоритма выгружаются из базы данных системы и подаются на вход модулю генерации подписи. По завершению работы алгоритма, результат



работы возвращается в систему и передается вызвавшему эту функцию пользователю.

Пример вывода работы в консоль алгоритма для документа с пятью версиями показаны на рисунке 8.

```
Document names: ['version_0.docx', 'version_1.docx', 'version_2.docx', 'version_3.docx', 'version_4.docx']
Message to be signed:
54d99ead440273463ce37e1b035d11119ea2e7fa21806aad3f1a28bdf6576c67
-----
secret key:
PrivateKey(0x69a609e4b6b94e360955eb04c3b2fe7a432fda70d1497a7685b2e65701815182)
private key:
JacobianPoint(x = Fq(0xcaa79..d31e7), y = Fq(0x18c80..e8e5d)z = Fq(0x12f2d..44dff))
signature:
JacobianPoint(x = Fq2(Fq(0xe978a..ad1c2), Fq(0x14e56..dddc5)), y = Fq2(Fq(0x4b1b1..3cd11), Fq(0xae0a..3e161)))z = Fq2(Fq(0x4e9b0..f3e7a), Fq(0x16f9c..1bcf9)))
time:
0.515
-----
secret key:
PrivateKey(0x33da1e8a2c3683e1a6fcd018dacd0d5522965feb008fa083a84ef12d094828a8)
private key:
JacobianPoint(x = Fq(0x1899e..df314), y = Fq(0x11a2f..6afb7)z = Fq(0x48426..0ee7a))
signature:
JacobianPoint(x = Fq2(Fq(0x120be..3a08d), Fq(0x19bbf..32770)), y = Fq2(Fq(0x69895..575c4), Fq(0x6eb80..42d84)))z = Fq2(Fq(0xf400a..dec82), Fq(0x51497..c2a4b)))
time:
0.5088
-----
secret key:
PrivateKey(0x7136997a78cbbc0e668c247d948b93e44469e61668e433e342c7fc43b7044ad0)
private key:
JacobianPoint(x = Fq(0x81cb6..3a96a), y = Fq(0x1030f..49947)z = Fq(0x38cdf..2c49d))
signature:
JacobianPoint(x = Fq2(Fq(0x49358..bbdf), Fq(0x4da20..291b6)), y = Fq2(Fq(0x193e8..0a8bd), Fq(0x17f86..73912)))z = Fq2(Fq(0xf19fb9..8811a), Fq(0x13e47..7abe4)))
time:
0.512
-----
secret key:
PrivateKey(0x5ae449e812ce51e1d5abcbdb00aa5b5308cad8306604d5bfff37a6bca8645af03)
private key:
JacobianPoint(x = Fq(0x90cc7..8758c), y = Fq(0x12681..ba69a)z = Fq(0x156f9..696ab))
signature:
JacobianPoint(x = Fq2(Fq(0x1462a..db3ab), Fq(0x328ff..08f18)), y = Fq2(Fq(0xf3b0c..997f9), Fq(0x125a2..b9425)))z = Fq2(Fq(0xe7813..b29df), Fq(0x1f172..7f834)))
time:
0.5199
-----
aggregate signature:
JacobianPoint(x = Fq2(Fq(0x6b12b..68641), Fq(0x133fb..008ec)), y = Fq2(Fq(0x2b47b..95036), Fq(0xd0d67..b3e18)))z = Fq2(Fq(0x78d3c..d3502), Fq(0xf77e2..3cd0d)))
1061555309980309121466849289927358934146936566642020594661522506399008146221244411848172333044091663421722986366143315525941337825498147895353518767414588508626175886137084733356785341940718335000327871287154626933448726287703459294
final time:
2.61
```

Рисунок 8 – Вывод работы программы в консоль

На вход подаются все версии документа, которые должны быть подписаны. Каждая версия документа, не считая финальной, является подписантом для этого алгоритма. Подписываемый документ переводится в хэш-сумму, с добавлением даты начала операции. После этого, хэш значение версий документов используется для создания секретных слов, которые далее используются для создания подписей. В конце работы алгоритма, все полученные подписи объединяются в агрегатную подпись, которая и является финальной подписью документа.

Пользователь получает сигнатуру в виде строки цифр, которые являются байтовым представлением точки на эллиптической кривой. Изображение того, что видит пользователь, показано на рисунке 9.

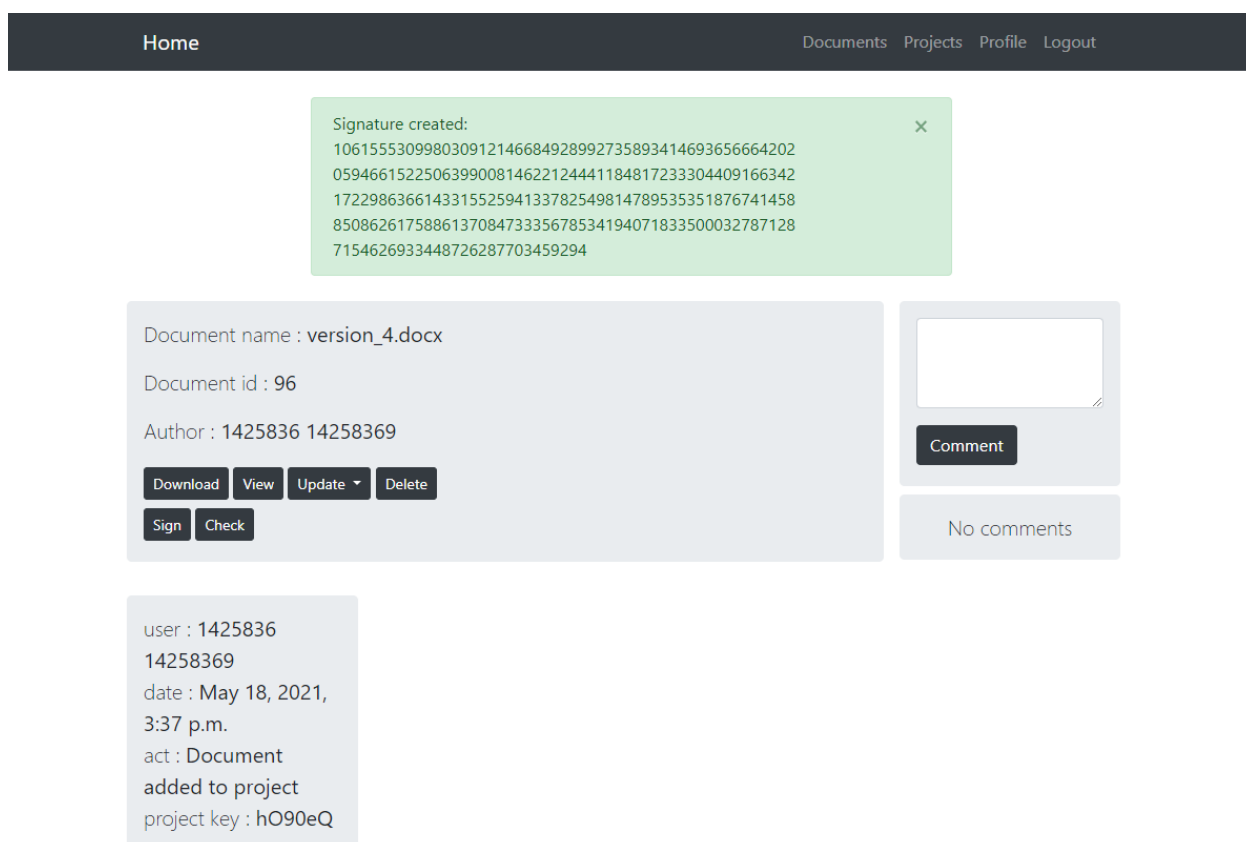


Рисунок 9 – Результат работы программы видимый для пользователя

### 3.3 Выводы

В ходе разбора аналогичных решений рынка систем электронного документооборота, были выявлены важные функциональные параметры системы. На основе этого было предложено техническое решение, а также создано техническое описание программного продукта. Было произведено внедрение в систему спроектированного алгоритма коллективной цифровой электронной подписи.

## 4 АНАЛИЗ ЭФФЕКТИВНОСТИ РАЗРАБОТАННОЙ СИСТЕМЫ

На рисунке 10 показаны этапы создания подписи для документа с четырьмя версиями.

```
Document names:
['version_0.docx', 'version_1.docx', 'version_2.docx', 'version_3.docx']
Message to be signed:
8149c061769adf0197f3d95a3d9212243643d54b536ebdce758a62f9c8883eae
-----
secret key:
PrivateKey(0x69a609e4b6b94e360955eb04c3b2fe7a432fda70d1497a7685b2e65701815182)
private key:
JacobianPoint(x = Fq(0xcaa79..d31e7), y = Fq(0x18c80..e8e5d))z = Fq(0x12f2d..44dff))
signature:
JacobianPoint(x = Fq2(Fq(0x2125d..5453b), Fq(0xd17a1..934b2)), y = Fq2(Fq(0xe73a3..3f189), Fq(0x12f14..39b04))z = Fq2(Fq(0x83329..e68f8), Fq(0xa4d22..129b0)))
time:
0.508
-----
secret key:
PrivateKey(0x33da1e8a2c3683e1a6fcd018dacd0d5522965feb008fa083a84ef12d094828a8)
private key:
JacobianPoint(x = Fq(0x1899e..df314), y = Fq(0x11a2f..6afb7))z = Fq(0x48426..0ee7a))
signature:
JacobianPoint(x = Fq2(Fq(0x15e95..cf06e), Fq(0xa9bab..e0df5)), y = Fq2(Fq(0xd02eb..edb8c), Fq(0x1261b..b24e8))z = Fq2(Fq(0x936a0..7036b), Fq(0xb9cd3..99fc0)))
time:
0.507
-----
secret key:
PrivateKey(0x7136997a78cbbc0e668c247d948b93e44469e61668e433e342c7fc43b7044ad0)
private key:
JacobianPoint(x = Fq(0x81cb6..3a96a), y = Fq(0x1030f..49947))z = Fq(0x38cdf..2c49d))
signature:
JacobianPoint(x = Fq2(Fq(0x71f96..f130c), Fq(0x3c5c2..2c342)), y = Fq2(Fq(0x140be..0c343), Fq(0x1564e..c4d31))z = Fq2(Fq(0x17796..b68bc), Fq(0x1077c..64a36)))
time:
0.5113
-----
aggregate signature:
JacobianPoint(x = Fq2(Fq(0xe8b96..30629), Fq(0x16f1f..9b03e)), y = Fq2(Fq(0xc3696..01d96), Fq(0x766e8..300bb))z = Fq2(Fq(0x181eb..3ad73), Fq(0x4425d..6856e)))
84547868362768391417226414755142935744508879300903113629186494679415600903992065651063193966715071118466008206283348530595471521147574816575022
7666147714599530038996115880952969195710705999985261759962274030277204900443547483447807
final time:
1.931
```

Рисунок 10 - этапы создания подписи

На вход функции подается список названий версий документа. Последнее имя считается финальной версией документ, которая и будет подписываться. На каждом этапе создается публичный и секретный ключ, и создается подпись документа. После создания всех индивидуальных подписей, они объединяются в одну, которая и является результатом работы алгоритма.

На рисунке 11 представлена зависимость количества экземпляров различных версий документа, ко времени затрачиваемому на создание подписи.

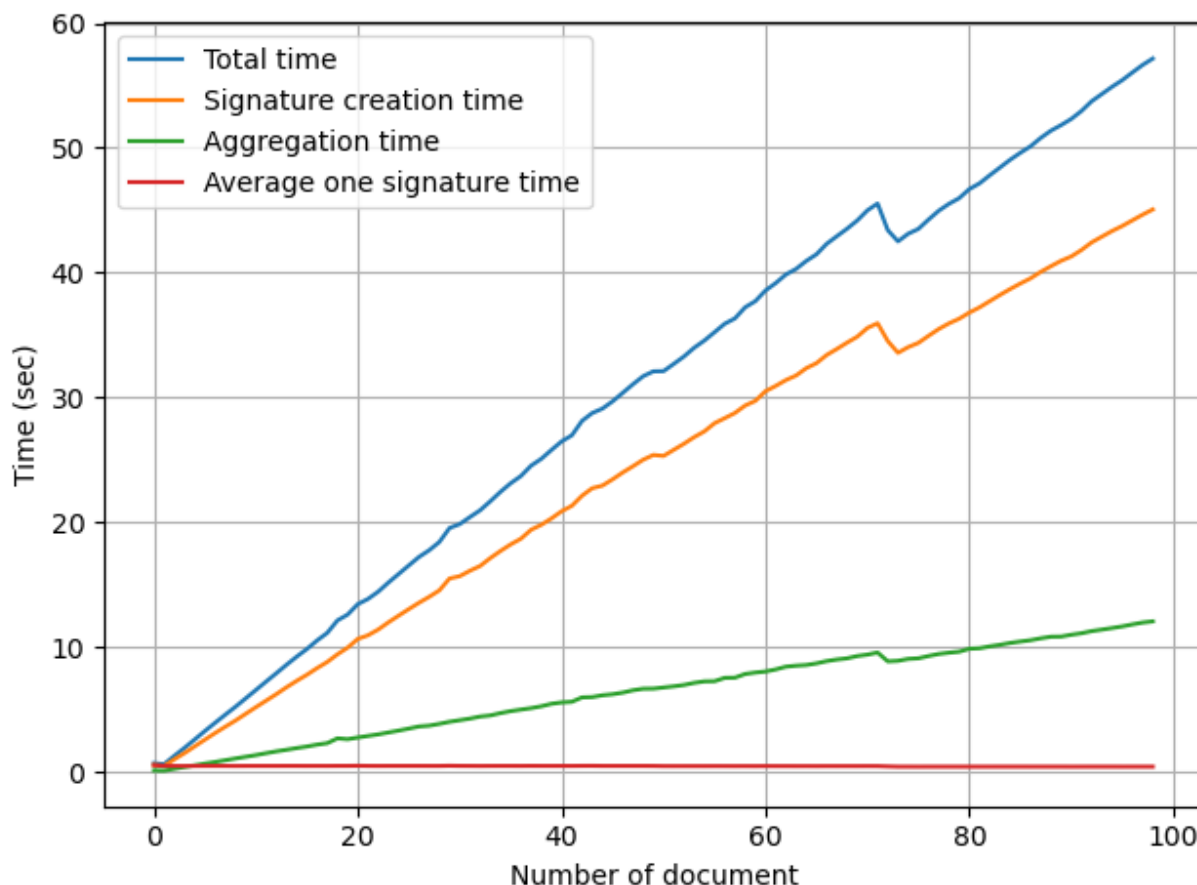


Рисунок 11 – Зависимость количества документов к затрачиваемому времени

Из представленного графика можно сделать вывод, что большую часть времени работы алгоритма занимает вычисление подписей для различных версий документа. Это вызвано сложностью вычисления точек на эллиптической кривой. Также, можно увидеть что среднее время, затрачиваемое на расчёт одной подписи, равняется половине секунды.

В качестве решения этой проблемы было решено использовать все ядра процессора. Результат работы улучшенного алгоритма представлен на рисунке 12.

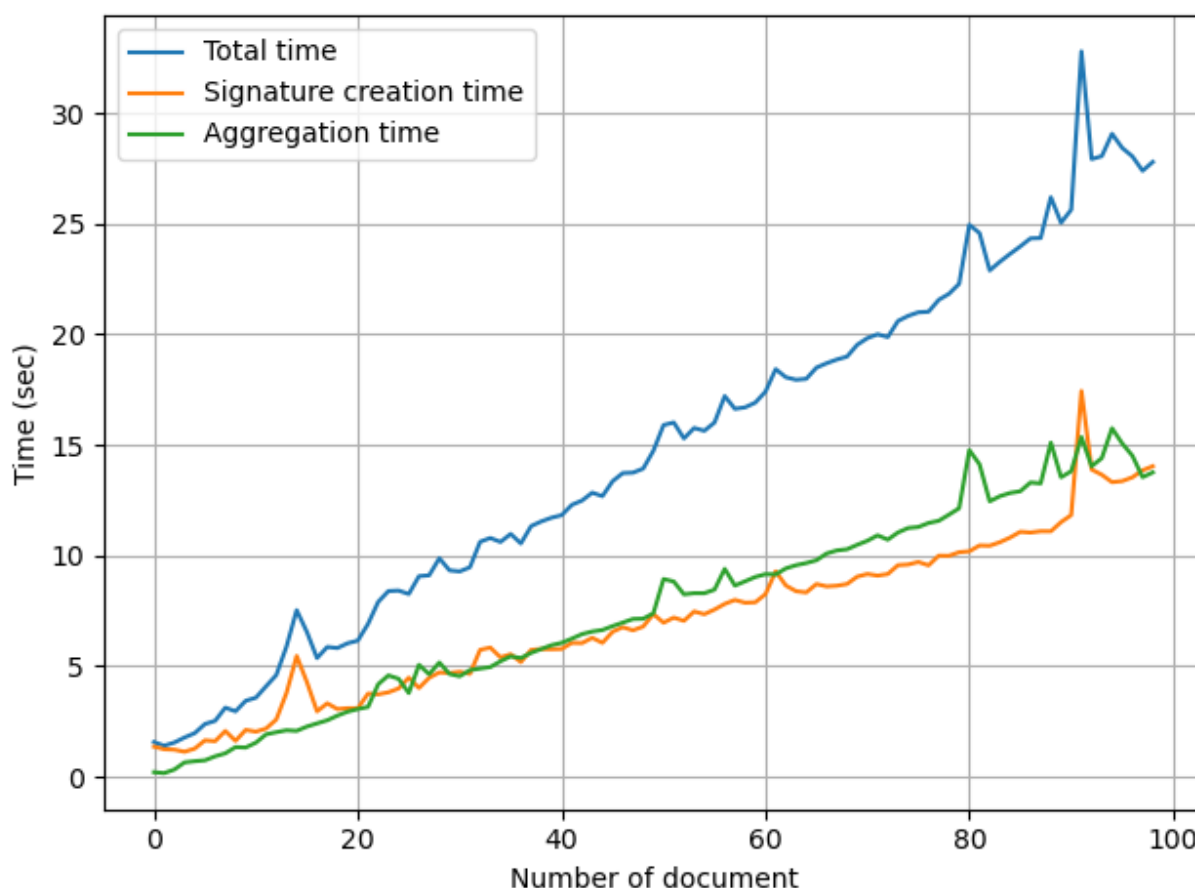


Рисунок 12 – Алгоритм подписи использующий процессорные ядра

Использование всех возможных ядер позволило увеличить скорость расчёта подписей в 3 – 4 раза, а общее время уменьшилось примерно в 2 раза. Флуктуации в графике могут быть вызваны множеством различных факторов. Скорее всего это связано с тем, что различные процессы занимают ядра процессора, и интерпретатору приходится тратить время на ожидание окончания предыдущего использования.

#### 4.1 Выводы

В ходе рассмотрения интегрированного алгоритма, был проведён анализ работоспособности системы на большом количестве данных, и выявлен существенный недостаток, заключающийся в низкой скорости создания подписи. Также, было предложено решение данной проблемы с помощью использования всех ядер процессора.

## ЗАКЛЮЧЕНИЕ

Целью данной работы являлось повышение уровня защищенности системы электронного документооборота с помощью алгоритма коллективной цифровой подписи без использования сертификационных центров.

В ходе выполнения выпускной квалификационной работы был изучен рынок систем электронного документооборота, различные алгоритмы создания электронной цифровой подписи, и на их основе был разработан алгоритм генерации коллективной цифровой подписи. Был произведён анализ аналогичных решений систем электронного документооборота, выявлены функциональные особенности и на их основе создано техническое описание разработанного приложения. Произведено внедрение алгоритма подписи в программный продукт, и проведены исследования на экспериментальном образце.

Результатом работы является система электронного документооборота с внедренным алгоритмом коллективной цифровой подписи.

Данная работа была представлена на X Всероссийском конгрессе молодых ученых. Работа была отмечена дипломом конкурса докладов для поступления в магистратуру Университета ИТМО.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Чернова А. Я. Анализ системы формирования и проверки электронной подписи. [Электронный ресурс]. – Электрон. дан. – Режим доступа: <https://cyberleninka.ru/article/n/analiz-sistemy-formirovaniya-i-proverki-elektronnoy-podpisi> [Дата обращения: 20.05.2021]
2. Encyclopedia by Kaspersky. Симметричное шифрование. [Электронный ресурс]. – Электрон. дан. – Режим доступа: <https://encyclopedia.kaspersky.ru/glossary/symmetric-encryption/> [Дата обращения: 20.05.2021]
3. Яндекс практикум. Асимметричное шифрование. [Электронный ресурс]. – Электрон. дан. – Режим доступа: <https://thecode.media/asymmetric/> [Дата обращения: 20.05.2021]
4. Ананьев М. Ю., Гортинская Л. В., Костин А. А., Молдовян Н. А. Вычислительная техника. Реализация протокола коллективной подписи на основе стандартов эцп. [Электронный ресурс]. – Электрон. дан. – Режим доступа: <https://pribor.ifmo.ru/file/article/4656.pdf> [Дата обращения: 20.05.2021]
5. Ротков Л.Ю., Зобнев А.В. Электронная цифровая подпись в электронном документообороте. [Электронный ресурс]. – Электрон. дан. – Режим доступа: <http://www.unn.ru/pages/e-library/aids/2006/19.pdf> [Дата обращения: 20.05.2021]
6. TAdviser. Обзор "Российский рынок СЭД/ЕСМ-систем". [Электронный ресурс]. – Электрон. дан. – Режим доступа: [https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%A1%D0%AD%D0%94\\_\(%D1%80%D1%8B%D0%BD%D0%BE%D0%BA\\_%D0%A0%D0%BE%D1%81%D1%81%D0%B8%D0%B8\)](https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%A1%D0%AD%D0%94_(%D1%80%D1%8B%D0%BD%D0%BE%D0%BA_%D0%A0%D0%BE%D1%81%D1%81%D0%B8%D0%B8)) [Дата обращения: 24.04.2021]
7. Тихонова М. А. Развитие моделей антикризисного управления малым бизнесом в условиях последствий пандемии. [Электронный ресурс]. –

- Электрон. дан. – Режим доступа:  
<https://mgimo.ru/upload/2020/11/tihonova-diss.pdf> [Дата обращения:  
20.05.2021]
8. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма. [Электронный ресурс]. – Электрон. дан. – Режим доступа:  
<https://docs.cntd.ru/document/1200004855> [Дата обращения:  
25.04.2021]
9. Процессы формирования и проверки электронной цифровой подписи. [Электронный ресурс]. – Электрон. дан. – Режим доступа:  
<https://docs.cntd.ru/document/1200026578> [Дата обращения:  
25.04.2021]
10. Синев В. Е. Методы построения и разработка практических протоколов групповой подписи и алгебраических алгоритмов защитных преобразований. [Электронный ресурс]. – Электрон. дан. – Режим доступа:  
[http://www.spiiras.nw.ru/dissovet/wp-content/uploads/2017/09/dissertacija\\_sinev.pdf](http://www.spiiras.nw.ru/dissovet/wp-content/uploads/2017/09/dissertacija_sinev.pdf) [Дата обращения:  
25.04.2021]
11. D. Boneh., S. Gorbunov., H. Wee, Z. Zhang. BLS Signature Scheme. [Электронный ресурс]. – Электрон. дан. – Режим доступа:  
<https://tools.ietf.org/html/draft-boneh-bls-signature-00>



## ПРИЛОЖЕНИЕ А

### Техническое описание

# 1 ОБЩИЕ СВЕДЕНИЯ О РАЗРАБОТКЕ

## 1.1 Наименование программы

Веб-сервис с системой распределения документов «DMS».

## 1.2 Языки программирования

Программный продукт (ПП) был разработан средствами языка программирования Python, фреймворка Django и фреймворка Bootstrap.

## 1.3 Назначение и функции, выполняемые программой

Назначением разработки является создание веб-сервиса с системой распределения документов.

В разработанном ПП выделены следующие функциональные возможности:

- возможность регистрации и авторизации пользователей;
- интерфейс для работы с документами и проектами;
- поиск документов;
- создание, редактирование или удаление документов;
- создание и проверка электронной подписи документов;
- создание, редактирование или удаление проектов;
- добавление, редактирование или удаление документов в проекте;
- логирование действий.

## 1.4 Описание особенностей программы

Для работы программы, на компьютере должна быть установлена операционная система поддерживающая язык программирования Python версии 3.8.3.

Должен быть установлен фреймворк Django версии 2.1.3 и пакеты:

- django-extensions 2.1.3;

– setuptools 40.6.2.

Для корректной работы модуля на стороне клиента должен быть установлен браузер, который содержит интерпретатор языка javascript, а также обрабатывает язык гипертекстовой разметки HTML 5 и каскадные таблицы стилей CSS. Работоспособность была проверена в браузерах Microsoft Edge версии 25.10586.672.0, Opera версии 45.0.2552.881, Internet Explorer версии 11.839.10586.0 и Google Chrome версии 58.0.3029.110.

## 2 СВЕДЕНИЯ О ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ

Входные и выходные данные представлены в таблице 1.

Таблица 1 - Входные и выходные данные.

Функция	Входные данные	Выходные данные
Регистрация на ресурсе	Адрес электронной почты, имя пользователя, пароль, имя и фамилия	Изменение информации в базе данных
Авторизация на ресурсе	Имя пользователя и пароль	Права доступа к функциям ресурса
Загрузка документа	Документ	Документ в библиотеке документов, изменение информации в базе данных
Удаление документа	Идентификатор документа	Удалённый документ, изменение информации в базе данных. Результат операции
Обновлять документ	Идентификатор документа	Обновленный документ в базе данных, логирование действия
Просматривать документ	Идентификатор документа	Документ
Скачивать документ на диск	Идентификатор документа	Документ
Создание электронной подписи	Идентификатор документа	Добавление открытого и закрытого ключа в базе данных
Проверять подпись	Идентификатор документа	Результат проверки
Комментировать документ	Идентификатор документа, текст	Добавление данных в базу данных. Вывод текста
Поиск документа	Идентификатор документа	Результат запроса

Продолжение таблицы 1

Просмотр профиля	Идентификатор пользователя	Информация о пользователе
Создание проекта	Имя проекта	Добавление данных в базу данных, логирование действия
Добавление пользователей в проект	Идентификатор пользователя	Изменение данных в базе данных
Просмотр информации о проекте	Идентификатор проекта	Результат запроса
Добавление документа в проект	Идентификатор документа и проекта	Изменение данных в базе данных, логирование действия
Комментировать проект	Идентификатор проекта, текст	Добавление данных в базу данных. Вывод текста
Удаление комментария	Идентификатор комментария, Идентификатор пользователя	Удаление данных из базы данных
Добавление задач	Идентификатор ответственного, дата дедлайна, текст	Добавление данных в базу данных
Выполнение задач	Идентификатор задачи	Изменение данных в базе данных

## 3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

### 3.1 Алгоритм программы

Программа разработана в соответствии с техническим заданием.

После запуска происходит инициализация основных исполняющих блоков: веб-сервера и базы данных.

После инициализации модулей, программа переходит в режим ожидания команд управления через веб-интерфейс. Приложение выполняет заданные функции в соответствии с входными данными, которые отправляет пользователь.

### 3.2 Структура программы, сведения о составных частях

Состав исходных файлов и распределение функций между ними представлен в таблице 2.

Таблица 2 - Описание исходных файлов программы

Файл	Назначение	Функции
manage.py	Загрузка настроек и запуск веб-сервера.	execute_from_command_line(sys.argv)
views.py	Рендеринг веб страниц и основная логика приложения.	index(request) login(request) logout(request) register(request) profile(request, user_id) edit_profile(request) change_password(request) create_new_document(user, file) documents(request)

Продолжение таблицы 2

		handle_uploaded_file(f, path) log(act, user, document = None, project = None) delete_document(request, doc_id) document(request, doc_id) download(request, doc_id) view(request, doc_id) RandomKeys(length) projects(request) add_document(request, project_key, document_id) delete_project(request, project_key) delete_from_project(request, project_key, document_id) project_log(request, project_key) sign_document(request, doc_id) check_document(request, doc_id) revert_log(request, doc_id, log_id)
settings.py	Настройки глобальных переменных.	-
models.py	Классы таблиц базы данных	-
forms.py	Классы форм ввода данных	-
admin.py	Настройки панели администратора	-

### 3.3 Описание выполнения функций

Функции системы реализованы в соответствии с техническим заданием. Ниже представлены изображения графических интерфейсов обеспечивающие функции системы.

Главное меню и пример оповещения показан на рисунке 1.

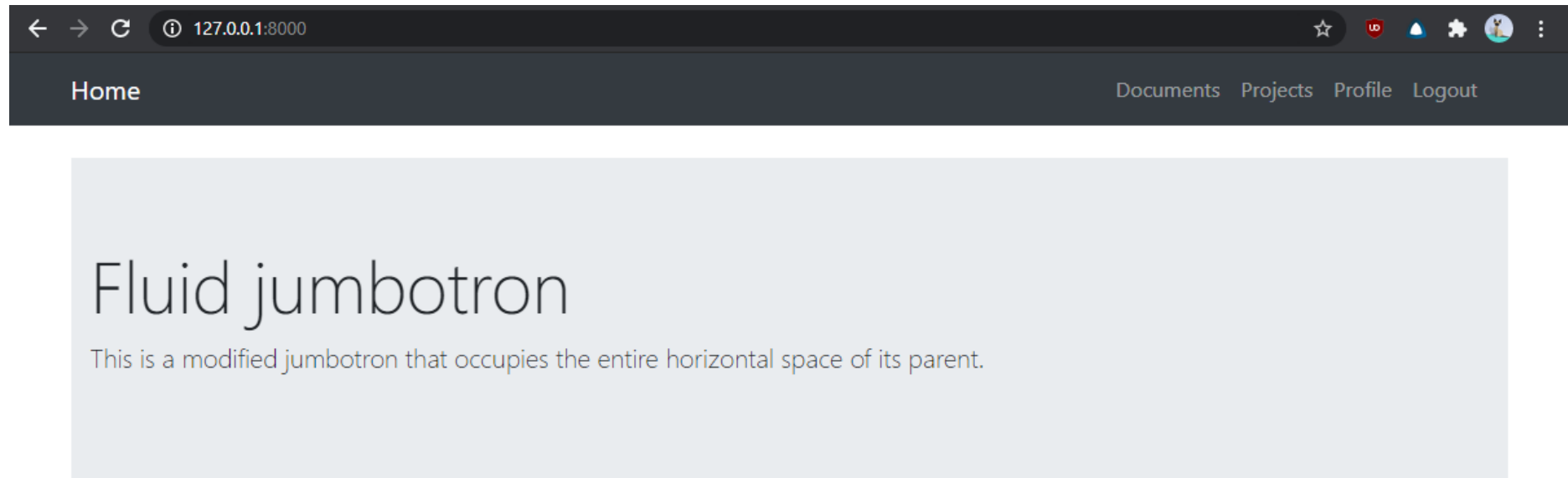


Рисунок 1 – Главное меню



Добавление документа показано на рисунке 2.

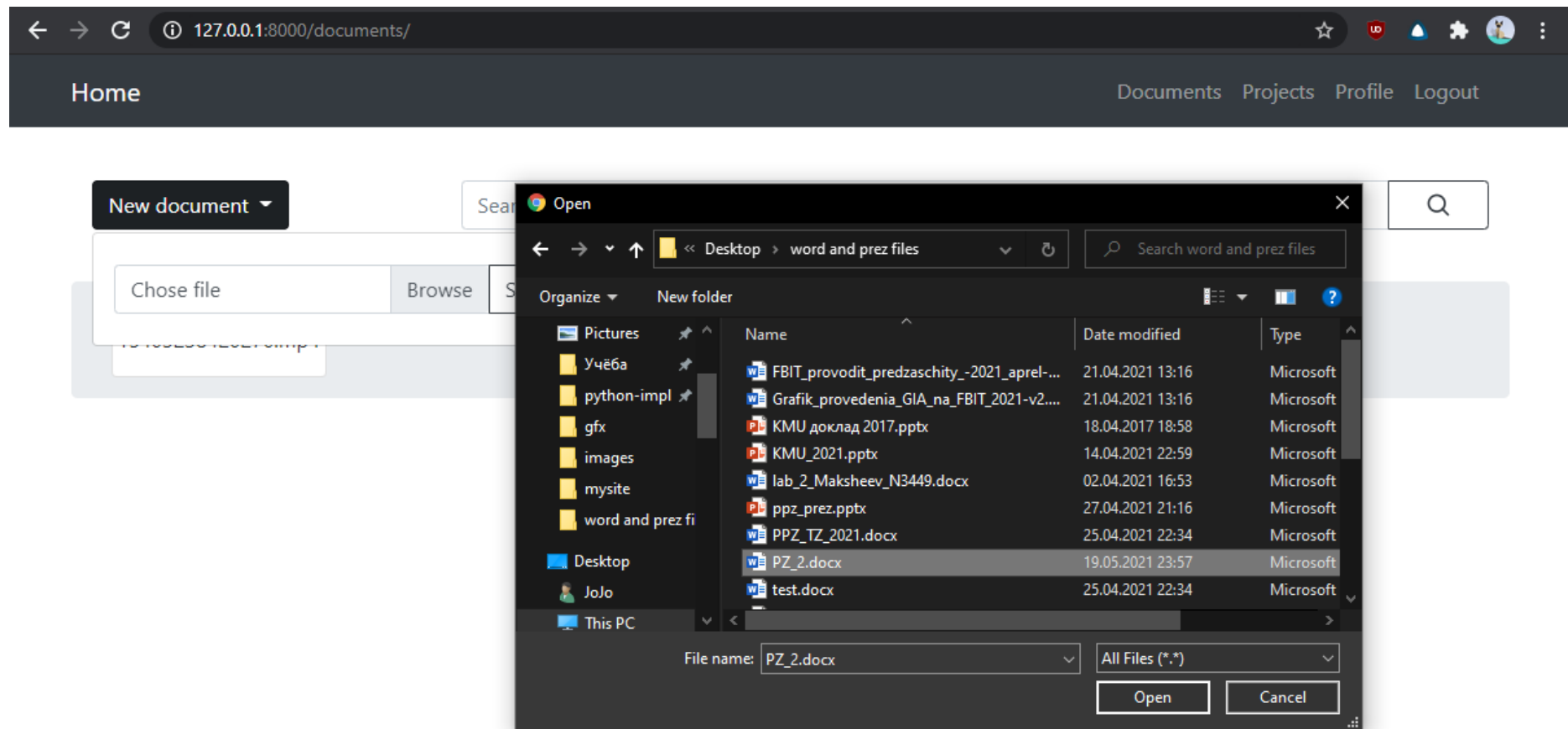


Рисунок 2 – Добавление документа

Добавленный документ показан на рисунке 3.

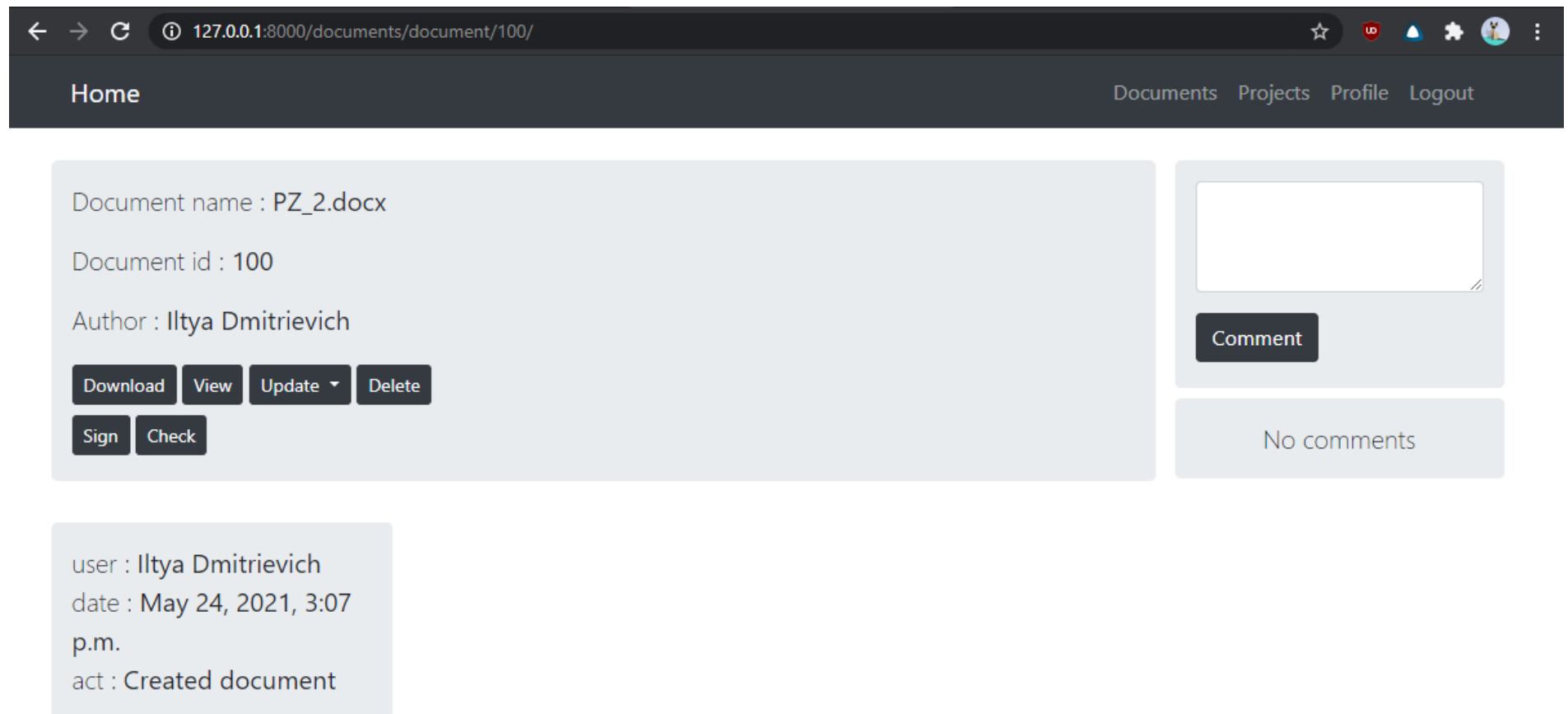


Рисунок 3 – Добавленный документ

Добавление комментария, обновление документа, подпись и проверка подписи показаны на рисунках 4 – 5.

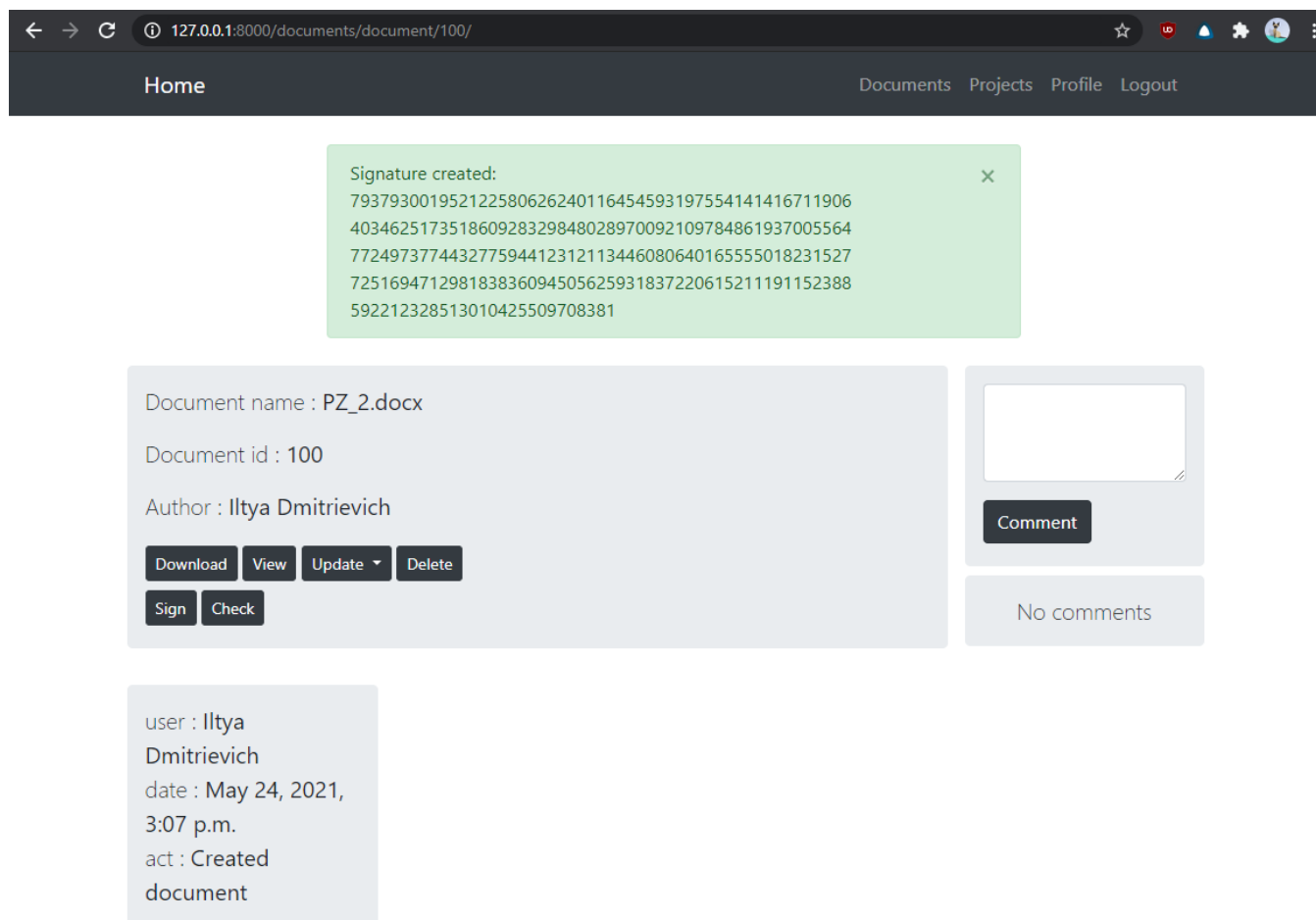


Рисунок 4 – Создание подписи

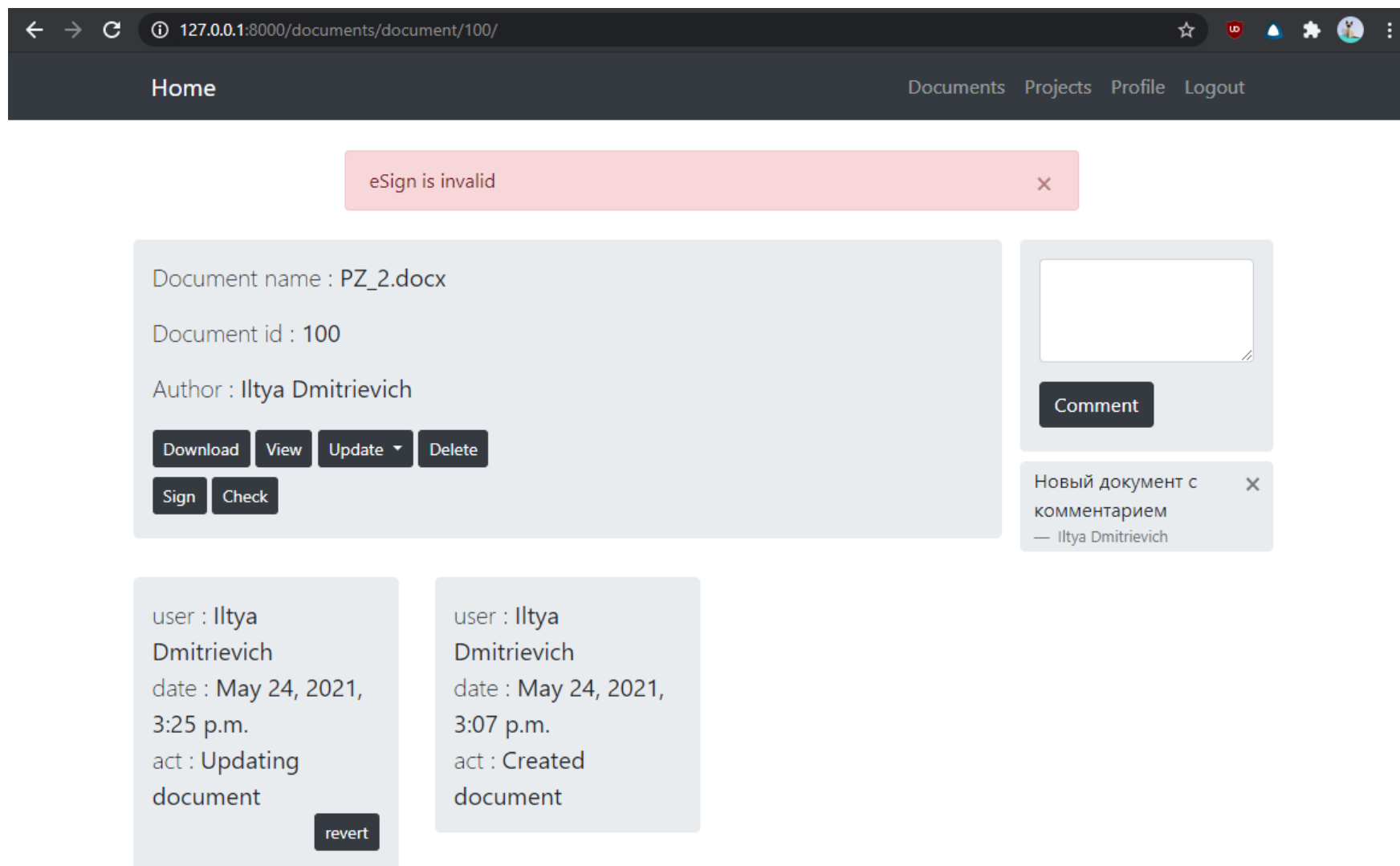


Рисунок 5 – Обновление документа, проверка подписи и добавленный комментарий

Функция добавление проекта показана на рисунке 6.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/projects/new\_project/'. The page has a dark header bar with 'Home' on the left and 'Documents', 'Projects', 'Profile', and 'Logout' on the right. Below the header, there is a section titled 'Users to add:' followed by a text input field containing the text 'asdfasdf asdfasdfasdf', 'Прикол Приколович', and 'Дмитрий Лагидзе'. Below this, there is a 'Project name:' label and an empty text input field. At the bottom left, there is a dark button labeled 'Create project'.

Рисунок 6 – Добавление проекта

Функция поиска документа рисунке 7.

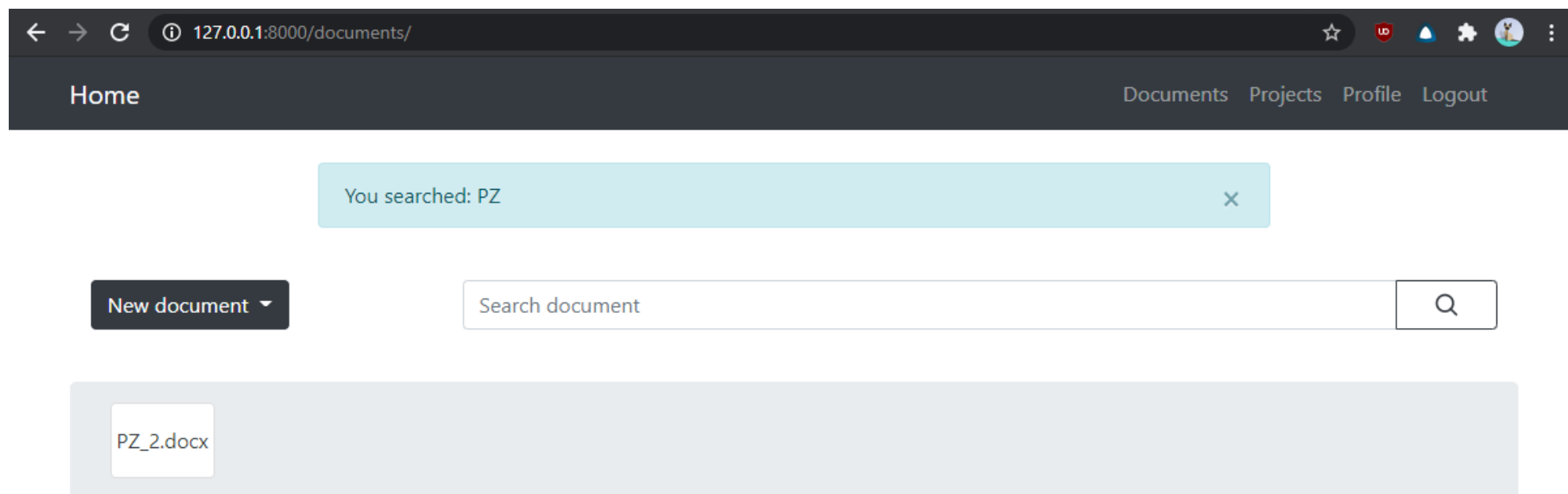


Рисунок 7 – Поиск документа

Функции добавления задач и документов в проекте показаны на рисунках 8 – 10.

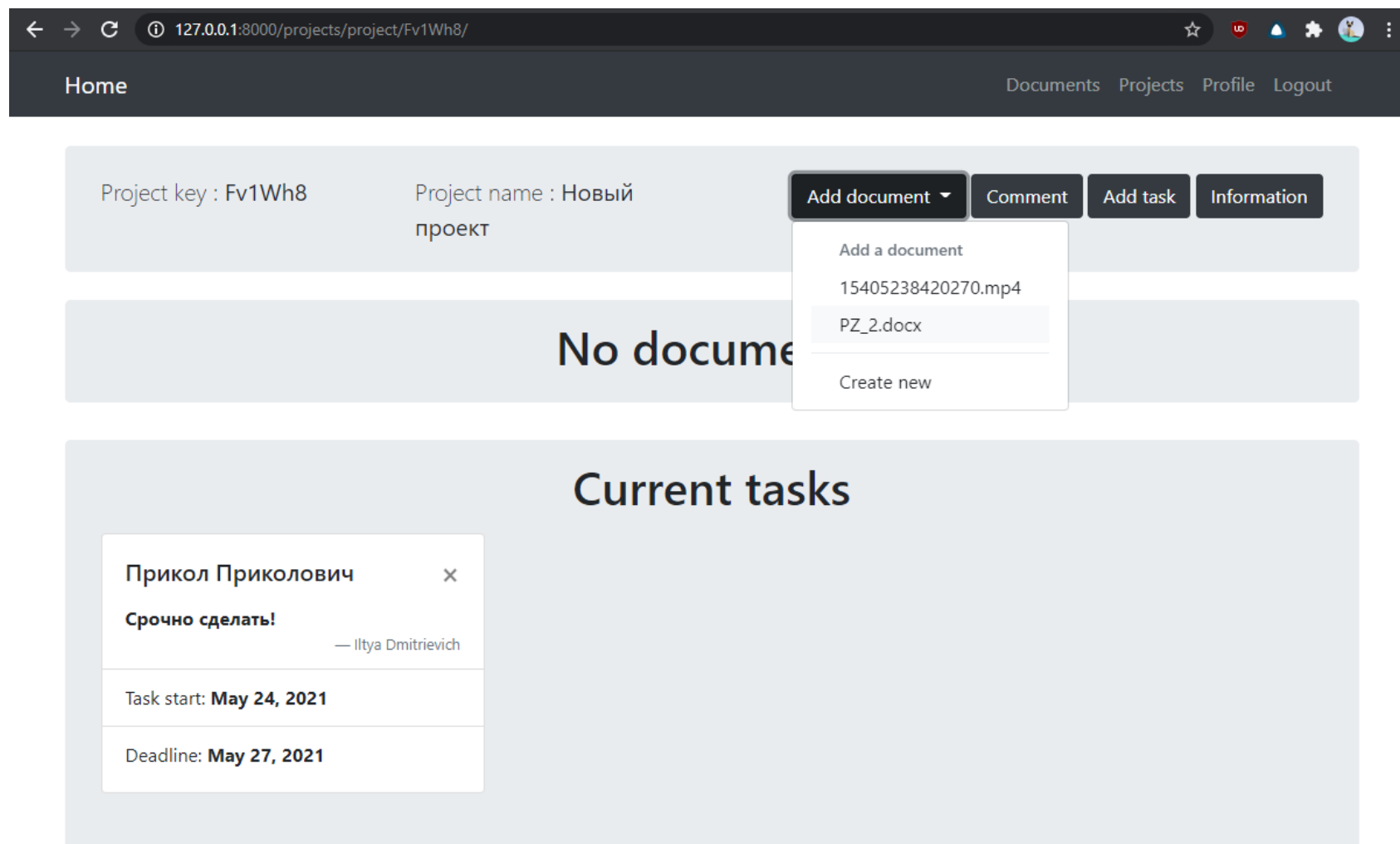


Рисунок 8 – Добавление документа в проект

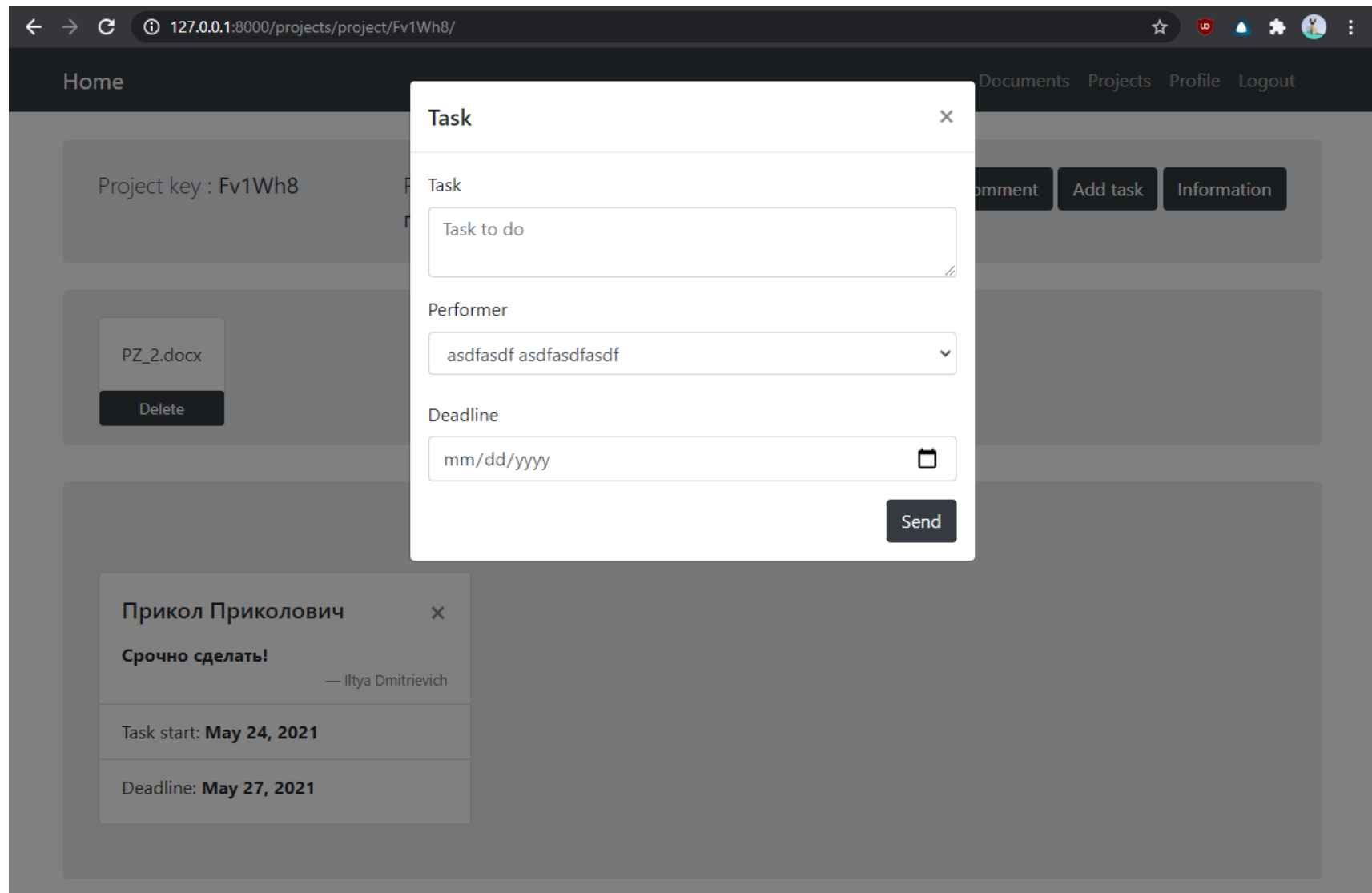


Рисунок 9 – Создание задачи



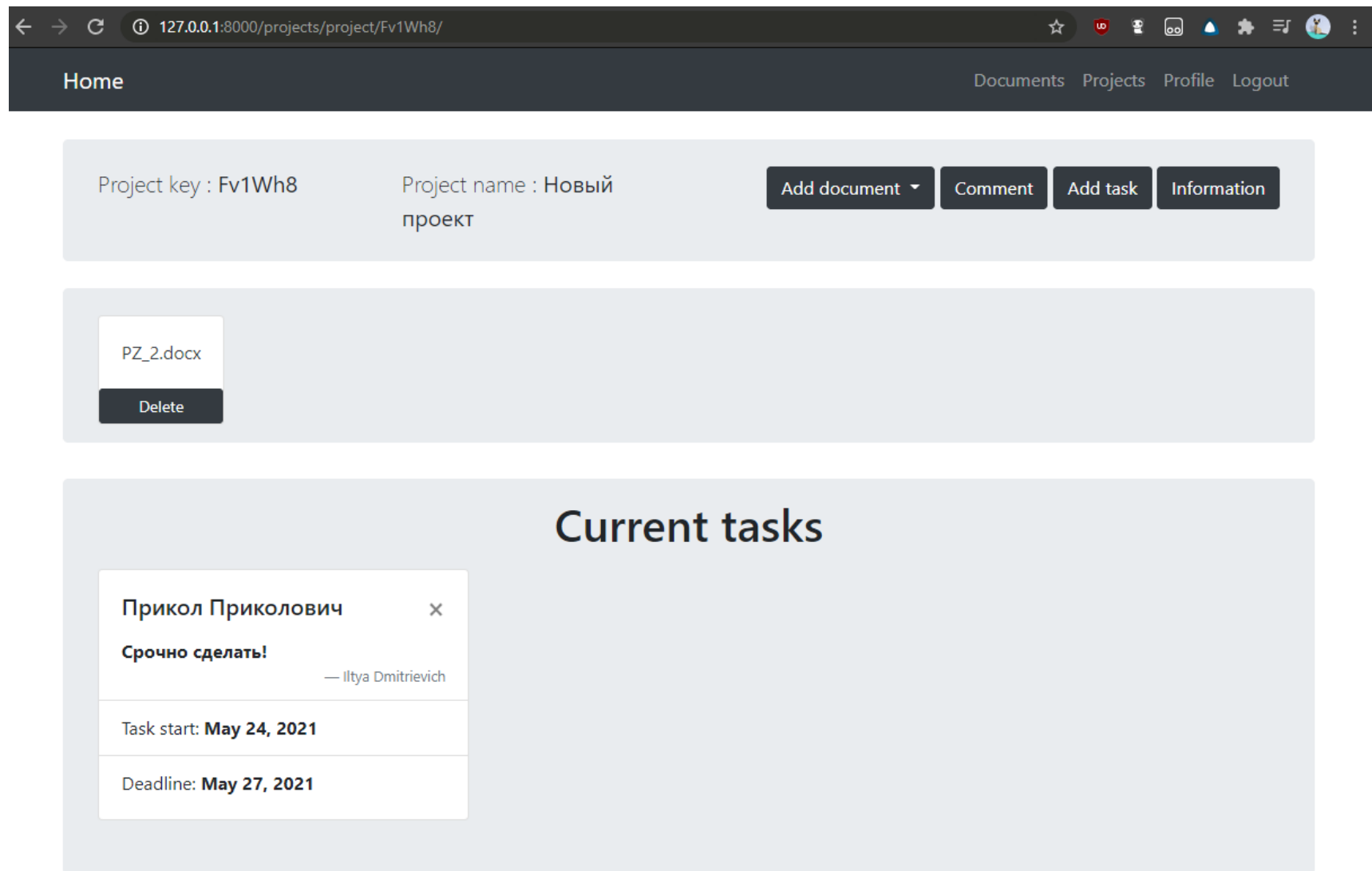


Рисунок 10 – Добавленный документ и задание в проекте

Информация о проекте и логирование действий показаны на рисунке 11.

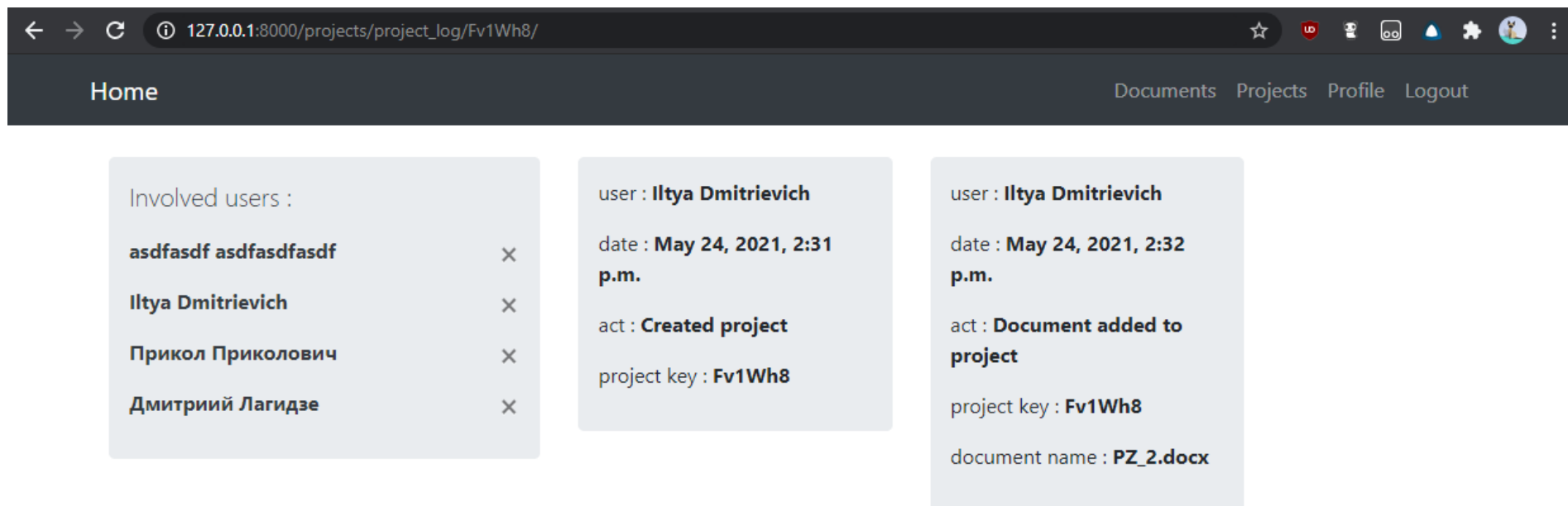


Рисунок 11 – Информация о проекте

## 4 ОПИСАНИЕ ВЫПОЛНЕНИЯ ПРОГРАММЫ

### 4.1 Загрузка программы

Установка программы не требуется. Программа размещена на сервере, который подключен к сети интернет.

### 4.2 Способы вызова программы

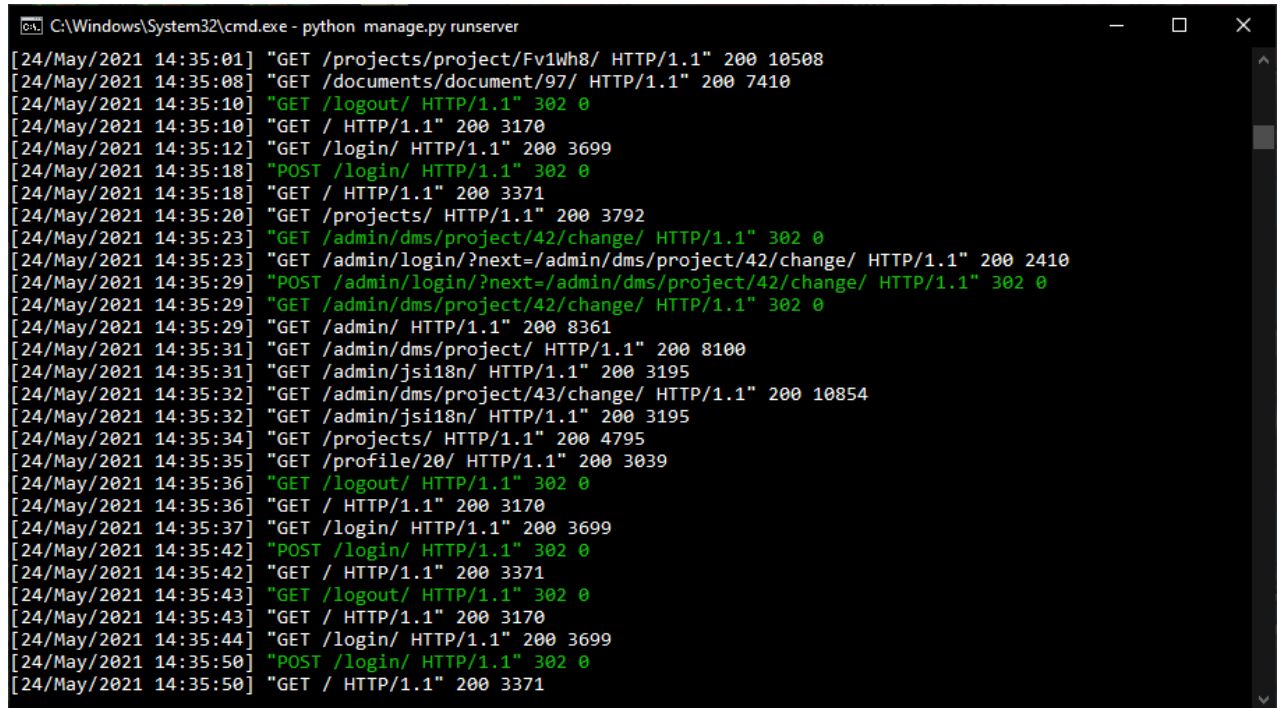
Использование приложения осуществляется путем указания адреса веб-ресурса в строке адреса программы-браузера.

### 4.3 Завершение выполнения программы

Завершение работы производится закрытием окна браузера (или вкладки), в котором выполнялась работа с веб-ресурсом.

## 5 СООБЩЕНИЯ ПРОГРАММЫ

Все сообщения о работе программы выводятся в консоль, где была запущена программа. Пример работы показан на рисунке 12.



```
C:\Windows\System32\cmd.exe - python manage.py runserver
[24/May/2021 14:35:01] "GET /projects/project/Fv1Wh8/ HTTP/1.1" 200 10508
[24/May/2021 14:35:08] "GET /documents/document/97/ HTTP/1.1" 200 7410
[24/May/2021 14:35:10] "GET /logout/ HTTP/1.1" 302 0
[24/May/2021 14:35:10] "GET / HTTP/1.1" 200 3170
[24/May/2021 14:35:12] "GET /login/ HTTP/1.1" 200 3699
[24/May/2021 14:35:18] "POST /login/ HTTP/1.1" 302 0
[24/May/2021 14:35:18] "GET / HTTP/1.1" 200 3371
[24/May/2021 14:35:20] "GET /projects/ HTTP/1.1" 200 3792
[24/May/2021 14:35:23] "GET /admin/dms/project/42/change/ HTTP/1.1" 302 0
[24/May/2021 14:35:23] "GET /admin/login/?next=/admin/dms/project/42/change/ HTTP/1.1" 200 2410
[24/May/2021 14:35:29] "POST /admin/login/?next=/admin/dms/project/42/change/ HTTP/1.1" 302 0
[24/May/2021 14:35:29] "GET /admin/dms/project/42/change/ HTTP/1.1" 302 0
[24/May/2021 14:35:29] "GET /admin/ HTTP/1.1" 200 8361
[24/May/2021 14:35:31] "GET /admin/dms/project/ HTTP/1.1" 200 8100
[24/May/2021 14:35:31] "GET /admin/jsi18n/ HTTP/1.1" 200 3195
[24/May/2021 14:35:32] "GET /admin/dms/project/43/change/ HTTP/1.1" 200 10854
[24/May/2021 14:35:32] "GET /admin/jsi18n/ HTTP/1.1" 200 3195
[24/May/2021 14:35:34] "GET /projects/ HTTP/1.1" 200 4795
[24/May/2021 14:35:35] "GET /profile/20/ HTTP/1.1" 200 3039
[24/May/2021 14:35:36] "GET /logout/ HTTP/1.1" 302 0
[24/May/2021 14:35:36] "GET / HTTP/1.1" 200 3170
[24/May/2021 14:35:37] "GET /login/ HTTP/1.1" 200 3699
[24/May/2021 14:35:42] "POST /login/ HTTP/1.1" 302 0
[24/May/2021 14:35:42] "GET / HTTP/1.1" 200 3371
[24/May/2021 14:35:43] "GET /logout/ HTTP/1.1" 302 0
[24/May/2021 14:35:43] "GET / HTTP/1.1" 200 3170
[24/May/2021 14:35:44] "GET /login/ HTTP/1.1" 200 3699
[24/May/2021 14:35:50] "POST /login/ HTTP/1.1" 302 0
[24/May/2021 14:35:50] "GET / HTTP/1.1" 200 3371
```

Рисунок 12 – Вывод программы