

# Отчет о проверке на заимствования №1



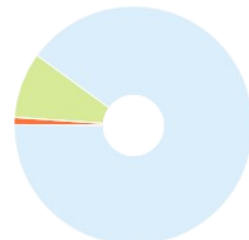
**Автор:** Носко Максим Михайлович  
**Проверяющий:** Пользователь для API (galkinala@fa.ru / ID: 2588)  
**Организация:** Финансовый университет при Правительстве РФ  
Отчет предоставлен сервисом «Антиплагиат» - <http://fa.antiplagiat.ru>

## ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 939776  
Начало загрузки: 03.06.2021 10:55:25  
Длительность загрузки: 00:00:13  
Имя исходного файла: ВКР Носко ПИ17-1.docx  
Название документа: ВКР Носко ПИ17-1.docx  
Размер текста: 1 кБ  
Символов в тексте: 106580  
Слов в тексте: 12981  
Число предложений: 503

## ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)  
Начало проверки: 03.06.2021 10:55:40  
Длительность проверки: 00:01:33  
Комментарии: не указано  
Поиск с учетом редактирования: да  
Модули поиска: ИПС Адилет, Библиография, Сводная коллекция ЭБС, Интернет Плюс, Сводная коллекция РГБ, Цитирование, Переводные заимствования (RuEn), Переводные заимствования по eLIBRARY.RU (EnRu), Переводные заимствования по Интернету (EnRu), Переводные заимствования издательства Wiley (RuEn), eLIBRARY.RU, Модуль поиска "ФУ", СПС ГАРАНТ, Медицина, Диссертации НББ, Перефразирования по eLIBRARY.RU, Перефразирования по Интернету, Патенты СССР, РФ, СНГ, Шаблонные фразы, Кольцо вузов, Издательство Wiley



### ЗАИМСТВОВАНИЯ

0,69%

### САМОЦИТИРОВАНИЯ

0%

### ЦИТИРОВАНИЯ

8,53%

### ОРИГИНАЛЬНОСТЬ

90,78%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.  
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.  
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.  
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.  
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.  
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.  
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.  
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Доля в тексте	Источник	Актуален на	Модуль поиска	Блоков в отчете	Блоков в тексте	Комментарии
[01]	7,96%	7,96%	не указано	раньше 2011	Библиография	1	1	
[02]	0,57%	0,57%	не указано	раньше 2011	Шаблонные фразы	19	19	
[03]	0%	0,29%	Красавина, Вера Александровна Современные тенденции и перспективы развития мирового рынка ИТ-услуг : диссертация ... кандидата экономических наук : 08.00.14 Москва 2020 <a href="http://dlib.rsl.ru">http://dlib.rsl.ru</a>	12 Янв 2021	Сводная коллекция РГБ	0	2	
[04]	0,27%	0,27%	Структура данных типа граф — курсовая работа <a href="http://turboreferat.ru">http://turboreferat.ru</a>	20 Мая 2020	Интернет Плюс	1	1	
[05]	0%	0,27%	Ответы На Фэпо Тестирование Английский Язык Бесплатно Рефераты <a href="https://skachatreferat.ru">https://skachatreferat.ru</a>	23 Апр 2021	Интернет Плюс	0	1	
[06]	0,24%	0,24%	Основы PowerShell: введение в объекты. <a href="http://elibrary.ru">http://elibrary.ru</a>	13 Янв 2013	Перефразирования по eLIBRARY.RU	1	1	
[07]	0%	0,21%	Создание веб-приложения для работы с файлами Jupyter Notebook в рамках учебного процесса <a href="https://yandex.ru">https://yandex.ru</a>	07 Мая 2021	Интернет Плюс	0	2	
[08]	0%	0,17%	Котляров, Александр Сергеевич Методы и средства поддержки высокоскоростных сетевых технологий на реконфигурируемых вычислителях : диссертация ... кандидата технических наук : 05.13.11 Ростов-на-Дону 2020 <a href="http://dlib.rsl.ru">http://dlib.rsl.ru</a>	12 Янв 2021	Сводная коллекция РГБ	0	1	
[09]	0,16%	0,16%	Излишне восторженный обзор. излишне восторженный обзор <a href="https://halzen.ru">https://halzen.ru</a>	04 Мая 2021	Интернет Плюс	3	3	

[10]	0%	0,16%	ОБРАБОТКИ ДАННЫХ ПО АДМИНИСТРАТИВНЫМ ПРАВОНАРУШЕНИЯМ В ОБЛАСТИ ДОРОЖНОГО ДВИЖЕНИЯ И БЛАГОУСТРОЙСТВА ТЕРРИТОРИЙ, ПОСТУПАЮЩИМ ОТ СПЕЦИАЛЬНЫХ ТЕХНИЧЕСКИХ СРЕДСТВ ФОТО-ВИДЕО ФИКСАЦИИ И СИСТЕМ ВЕСОВОГО КОНТРОЛЯ ТРАНСПОРТНЫХ СРЕДСТВ <a href="http://fvf-rbs.ru">http://fvf-rbs.ru</a>	04 Мая 2021	Интернет Плюс	0	3	
[11]	0,01%	0,15%	Налоги и налоговая грамотность: сборник научных работ <a href="https://e.lanbook.com">https://e.lanbook.com</a>	22 Янв 2020	Сводная коллекция ЭБС	1	1	
[12]	0%	0,15%	Фискальная политика как инструмент преодоления современных барьеров экономического развития России: сборник научных статей <a href="https://e.lanbook.com">https://e.lanbook.com</a>	22 Янв 2020	Сводная коллекция ЭБС	0	1	
[13]	0%	0,14%	Файл публикации (формат Adobe PDF, размер 1751 Кб) » <a href="http://ict.informika.ru">http://ict.informika.ru</a>	29 Окт 2017	Интернет Плюс	0	2	Источник исключен. Причина: Маленький процент пересечения.
[14]	0%	0,14%	Перспективные направления развития банковского дела: Т.2 <a href="https://book.ru">https://book.ru</a>	03 Июл 2017	Сводная коллекция ЭБС	0	1	Источник исключен. Причина: Маленький процент пересечения.
[15]	0%	0,14%	Менеджмент и Бизнес-Администрирование: научно-практическое издание. 2017. № 2 <a href="http://biblioclub.ru">http://biblioclub.ru</a>	21 Янв 2020	Сводная коллекция ЭБС	0	1	Источник исключен. Причина: Маленький процент пересечения.
[16]	0%	0,14%	61447 <a href="http://e.lanbook.com">http://e.lanbook.com</a>	09 Мар 2016	Сводная коллекция ЭБС	0	1	Источник исключен. Причина: Маленький процент пересечения.
[17]	0%	0,14%	Управление прибылью в акционерных обществах региона: теория и практика. Книга 1 <a href="http://studentlibrary.ru">http://studentlibrary.ru</a>	19 Дек 2016	Медицина	0	1	Источник исключен. Причина: Маленький процент пересечения.
[18]	0%	0,14%	Сборник официальных документов и материалов 4/2016 <a href="http://studentlibrary.ru">http://studentlibrary.ru</a>	20 Дек 2016	Медицина	0	1	Источник исключен. Причина: Маленький процент пересечения.
[19]	0%	0,14%	Новости высоких технологий <a href="https://ixbt.com">https://ixbt.com</a>	07 Мая 2021	Интернет Плюс	0	3	Источник исключен. Причина: Маленький процент пересечения.
[20]	0%	0,13%	Трансформация банка в условиях глобальной неустойчивой среды <a href="https://unecon.ru">https://unecon.ru</a>	11 Июл 2020	Интернет Плюс	0	2	Источник исключен. Причина: Маленький процент пересечения.
[21]	0%	0,12%	HUAWEI MateBook D 14   HUAWEI Россия <a href="https://consumer.huawei.com">https://consumer.huawei.com</a>	07 Мая 2021	Интернет Плюс	0	2	Источник исключен. Причина: Маленький процент пересечения.
[22]	0%	0,12%	Совершенствование бизнес-процесса регистрации заявок в системе «ITSM» на примере (ООО ЧОО «Родина») - Бизнес планирование, Дипломная работа <a href="https://prepod24.ru">https://prepod24.ru</a>	05 Июл 2020	Интернет Плюс	0	2	Источник исключен. Причина: Маленький процент пересечения.
[23]	0%	0,11%	«НОВОЕ КАЧЕСТВО СОЦИАЛЬНО-ЭКОНОМИЧЕСКОГО ПРОГРЕССА: ВЫЗОВЫ И ШАНСЫ» (1/2) <a href="http://fa.ru">http://fa.ru</a>	20 Авг 2019	Интернет Плюс	0	2	Источник исключен. Причина: Маленький процент пересечения.
[24]	0%	0,11%	ОРГАНИЗАЦИЯ КОНТРОЛЯ ЗА ПРЕДПРИЯТИЯМИ ЖИЛИЩНО-КОММУНАЛЬНОЙ СФЕРЫ. <a href="http://elibrary.ru">http://elibrary.ru</a>	01 Фев 2021	eLIBRARY.RU	0	2	Источник исключен. Причина: Маленький процент пересечения.
[25]	0%	0,11%	pankratov_v_d_razrabotka-programmnogo-modulya-avtomatizacii-obzvona-klientov.pdf	15 Мая 2017	Кольцо вузов	0	1	Источник исключен. Причина: Маленький процент пересечения.
[26]	0%	0,1%	Оценка факторов риска и производственного стресса у разработчиков ИТ-продуктов. <a href="http://elibrary.ru">http://elibrary.ru</a>	18 Мая 2021	eLIBRARY.RU	0	2	Источник исключен. Причина: Маленький процент пересечения.
[27]	0%	0,09%	Содержание журнала "НАУЧНЫЕ ЗАПИСКИ МОЛОДЫХ ИССЛЕДОВАТЕЛЕЙ" за 2018 г. <a href="http://elibrary.ru">http://elibrary.ru</a>	23 Янв 2020	Перефразирования по eLIBRARY.RU	0	1	Источник исключен. Причина: Маленький процент пересечения.
[28]	0%	0,09%	tuzinec_v_a_sobstvennye-media-kak-sposob-b2c-kommunikacii-v-bankovskoy-sfere-na-rossiyskom-rynke.docx	21 Мая 2020	Кольцо вузов	0	1	Источник исключен. Причина: Маленький процент пересечения.
[29]	0%	0,09%	32154321.docx	20 Июн 2020	Кольцо вузов	0	1	Источник исключен. Причина: Маленький процент пересечения.
[30]	0%	0,09%	Шумская 1.docx	20 Июн 2020	Кольцо вузов	0	1	Источник исключен. Причина: Маленький процент пересечения.
[31]	0%	0,09%	Циркуляционная экономика и устойчивое развитие передовой мировой опыт.docx	20 Июн 2020	Кольцо вузов	0	1	Источник исключен. Причина: Маленький процент пересечения.

[32]	<div><div></div></div> 0%	0,09%	Google Новости <a href="https://news.google.com">https://news.google.com</a>	04 Мая 2021	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[33]	<div><div></div></div> 0%	0,08%	Объем ОЗУ и емкость аккумуляторов iPhone 11, iPhone 11 Pro и iPhone 11 Pro Max <a href="https://ixbt.com">https://ixbt.com</a>	04 Мая 2021	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[34]	<div><div></div></div> 0%	0,08%	Объем ОЗУ и емкость аккумуляторов iPhone 11, iPhone 11 Pro и iPhone 11 Pro Max <a href="https://ixbt.com">https://ixbt.com</a>	04 Мая 2021	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[35]	<div><div></div></div> 0%	0,07%	Firefox <a href="http://en.wikipedia.org">http://en.wikipedia.org</a>	04 Мая 2021	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[36]	<div><div></div></div> 0%	0,07%	Мобильный Firefox — Википедия <a href="https://ru.wikipedia.org">https://ru.wikipedia.org</a>	04 Мая 2021	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[37]	<div><div></div></div> 0%	0,07%	Кафедраның тәрбие жұмысына бағытталған мобильді қосымшаның сервері   Скачать Дипломдық жұмыс <a href="https://stud.kz">https://stud.kz</a>	07 Мая 2021	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[38]	<div><div></div></div> 0%	0,07%	Пожилые и стареющий социум России: выбор модели жизнедеятельности <a href="http://studentlibrary.ru">http://studentlibrary.ru</a>	19 Дек 2016	Медицина	0	1	Источник исключен. Причина: Маленький процент пересечения.
[39]	<div><div></div></div> 0%	0,07%	<a href="https://www.rsuh.ru/upload/main/vestnik/eup/Vestnik_eup4%20(14)_2018.pdf">https://www.rsuh.ru/upload/main/vestnik/eup/Vestnik_eup4%20(14)_2018.pdf</a> <a href="https://rsuh.ru">https://rsuh.ru</a>	08 Окт 2020	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[40]	<div><div></div></div> 0%	0,07%	Гальпер, Мария Андреевна Трансформация банка в условиях глобальной неустойчивой среды : диссертация ... кандидата экономических наук : 08.00.10 Санкт-Петербург 2020 <a href="http://dlib.rsl.ru">http://dlib.rsl.ru</a>	07 Сен 2020	Сводная коллекция РГБ	0	1	Источник исключен. Причина: Маленький процент пересечения.
[41]	<div><div></div></div> 0%	0,06%	Виды программного обеспечения и их классификация. Операционная система — allRefs.net <a href="http://allrefs.net">http://allrefs.net</a>	04 Мая 2021	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[42]	<div><div></div></div> 0%	0,06%	Исследование процессов "цифровизации" банковского сектора в рамках экосистемы цифровой экономики России <a href="https://revolution.allbest.ru">https://revolution.allbest.ru</a>	20 Мая 2020	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[43]	<div><div></div></div> 0%	0,06%	МСБ-кредитование: проблемы традиционного сектора и перспективы онлайн сервисов   Статья в журнале «Молодой ученый» <a href="https://moluch.ru">https://moluch.ru</a>	17 Янв 2021	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[44]	<div><div></div></div> 0%	0,06%	Банковское дело процесс цифровизации 2019 год <a href="https://mos-advokatam.ru">https://mos-advokatam.ru</a>	15 Окт 2020	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[45]	<div><div></div></div> 0%	0,06%	<a href="https://poisk.ivgpu.com/wp-content/uploads/2020/04/%D0%9F%D0%9E%D0%98%D0%A1%D0%9A-2020-27.04-2.pdf">https://poisk.ivgpu.com/wp-content/uploads/2020/04/%D0%9F%D0%9E%D0%98%D0%A1%D0%9A-2020-27.04-2.pdf</a> <a href="https://poisk.ivgpu.com">https://poisk.ivgpu.com</a>	02 Окт 2020	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[46]	<div><div></div></div> 0%	0,05%	<a href="http://mnic-penza.ru/inform/conf/sb_vk-22-19-2.pdf">http://mnic-penza.ru/inform/conf/sb_vk-22-19-2.pdf</a> <a href="http://mnic-penza.ru">http://mnic-penza.ru</a>	20 Июн 2019	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[47]	<div><div></div></div> 0%	0,05%	Диссертация на тему «Социокультурные факторы формирования кредитного поведения населения: региональный аспект», скачать бесплатно автореферат по специальности ВАК РФ 22.00.03 - Экономическая социология и демография <a href="https://dissercat.com">https://dissercat.com</a>	17 Мая 2021	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[48]	<div><div></div></div> 0%	0,05%	Александров А.С., Андреева О.И., Зайцев О.А. О перспективах развития российского уголовного судопроизводства в условиях цифровизации   Международная ассоциация содействия правосудию (МАСП/IAJ) <a href="https://iajaj.net">https://iajaj.net</a>	15 Окт 2020	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[49]	<div><div></div></div> 0%	0,05%	<a href="http://ineconomic.ru/sites//field_print_version/ineconomic-5-53-3-new.pdf">http://ineconomic.ru/sites//field_print_version/ineconomic-5-53-3-new.pdf</a> <a href="http://ineconomic.ru">http://ineconomic.ru</a>	25 Июн 2020	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.

2  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

2  
«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ  
РОССИЙСКОЙ ФЕДЕРАЦИИ»

11  
Департамент анализа данных и машинного обучения

Факультет информационных технологий и анализа больших данных

Выпускная квалификационная работа

на тему:

2  
Разработка ПО для расширения полномочий доверенных веб-страниц

Выполнил студент группы ПИ17-1

Носко Максим Михайлович \_\_\_\_\_

Научный руководитель

к.э.н., доцент

Коротеев Михаил Викторович \_\_\_\_\_

**ВКР соответствует предъявленным  
требованиям**

Руководитель департамента

д.э.н., профессор

\_\_\_\_\_ В. И. Соловьев

«\_\_\_» \_\_\_\_\_ 20\_\_ г.

Москва, 2021

## ОГЛАВЛЕНИЕ

<u>ВВЕДЕНИЕ</u> .....	5
<u>ГЛАВА 1. ИССЛЕДОВАНИЕ ВЗАИМОДЕЙСТВИЯ ВЕБ-СТРАНИЦ С УСТРОЙСТВОМ ПОЛЬЗОВАТЕЛЯ</u> .....	6
<u>1.1. Исследование текущей ситуации во взаимодействии веб-страниц с устройством пользователя</u> .....	6
<u>1.1.1. Исследование взаимодействия устройства и веб-страниц</u> .....	6
<u>1.1.2. Исследование популярных устройств и их характеристик</u> .....	7
<u>1.2. Исследование основных тенденций во взаимодействии веб-страниц с устройством пользователя</u> .....	13
<u>1.2.1. Исследование тенденций в предоставляемом пользователям ПО среди популярных сервисов</u> .....	13
<u>1.2.2. Исследование основных тенденций, связанных с полномочиями веб-страниц в популярных браузерах</u> .....	17
<u>ГЛАВА 2. ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ РАСШИРЕНИЯ ПОЛНОМОЧИЙ ДОВЕРЕННЫХ ВЕБ-СТРАНИЦ</u> .....	20
<u>2.1. Исследование возможности и способов расширения полномочий веб-страниц</u> .....	20
<u>2.1.1. Исследование алгоритма взаимодействия веб-страницы и пользователя</u> .....	20
<u>2.1.2. Исследование возможности расширения полномочий веб-страниц с помощью специализированного браузера</u> .....	22
<u>2.1.3. Исследование возможности расширения полномочий веб-страниц с помощью дополнительного ПО и браузера</u> .....	24
<u>2.1.4. Определение списка необходимых операций, выполняемых дополнительными ПО для расширения полномочий веб-страниц</u> .....	28

2.1.5. Составление алгоритма работы ПО для расширения полномочий веб-страниц.....	31
2.2. Исследование возможности и способов определения уровня доверия к содержимому веб-страницы.....	33
2.2.1. Исследование возможности использования разработанного алгоритма для определения уровня доверия к содержимому веб-страницы.....	33
2.2.2. Определение перечня дополнительных компонентов ПО для определения уровня доверия к содержимому веб-страницы.....	35
2.2.3. Составление алгоритма работы ПО для расширения полномочий доверенных веб-страниц.....	36
Описанная выше архитектура позволяет обеспечить расширение полномочий доверенных веб-страниц, в том числе, с учётом обеспечения проверки приложения на предмет доверия.....	39
2.3. Проектирование программного обеспечения.....	39
2.3.1. Проектирование перечня компонентов системы расширения полномочий доверенных веб-страниц и технологий их реализации.....	39
2.3.2. Определение полномочий и задач пользователей системы.....	43
2.3.3. Формирование требований к системе.....	45
ГЛАВА 3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	47
3.1. Разработка программного обеспечения для определения уровня доверия к содержимому веб-страницы.....	47
3.1.1. Проектирование базы данных СРП.....	47
3.1.2. Разработка Сервера СРП.....	48
3.1.3. Разработка Портала СРП.....	56
3.1.4. Разработка Серверной библиотеки СРП.....	67

<u>3.2. Разработка программного обеспечения для расширения полномочий доверенных веб-страниц.....</u>	68
<u>3.2.1. Разработка Клиента СРП.....</u>	68
<u>3.2.2. Разработка Клиентской библиотеки СРП.....</u>	69
<u>3.3. Разработка набора веб-приложений, использующих программное обеспечение для расширения полномочий доверенных веб-страниц.....</u>	71
<u>ЗАКЛЮЧЕНИЕ.....</u>	74
<u>СПИСОК ЛИТЕРАТУРЫ.....</u>	75
<u>ПРИЛОЖЕНИЕ. ИСХОДНЫЙ КОД ПРОГРАММЫ.....</u>	81

История развития интернета неразрывно связана с историей развития веб-технологий, и, соответственно, с развитием веб-браузеров, задачей которых всегда было наиболее полноценное и интерактивное отображение информации в браузере, совмещённое с максимальной экономией объёма пересылаемых по компьютерным сетям информации.

Развитие интерактивности веб-страниц привело к существованию веб-приложений, то есть веб-страниц, в которых, во многих случаях, ценность для пользователя представляет не контент, который хранится на сервере, а эффективность выполнения задачи пользователя, как затрагивающей другие устройства или людей, так и локальной задачи, связанной только с пользователем и его текущим устройством, на котором открыта веб-страница.

Такая интерактивность стала возможной благодаря тому, что браузеры стали позволять странице иметь, помимо текста и простейших не интерактивных, но дополнительно загружаемых элементов, ещё и исходный код программы, которая будет выполнена браузером при открытии страницы.

У интерактивности веб-страниц уже произошло несколько этапов развития, на каждом из которых, вместе с новыми возможностями, появлялись новые угрозы устройству пользователей. Разумеется, на каждом этапе данного развития, ведущие веб-браузеры внедряли средства контроля, которые позволяли пресекать нежелательные действия со стороны веб-страницы по отношению к устройству пользователя.

Но до сих пор веб-страницы уступают устанавливаемым приложениям по многим полномочиям, что связано с недостаточным уровнем обеспечения безопасности при использовании данных полномочий со стороны веб-браузеров.



## ГЛАВА 1. ИССЛЕДОВАНИЕ ВЗАИМОДЕЙСТВИЯ ВЕБ-СТРАНИЦ С УСТРОЙСТВОМ ПОЛЬЗОВАТЕЛЯ

### 1.1. Исследование текущей ситуации во взаимодействии веб-страниц с устройством пользователя

#### 1.1.1. Исследование взаимодействия устройства и веб-страниц

Взаимодействие веб-страницы с устройством пользователя определяется, прежде всего, браузером, так как сама по себе веб-страница, вне браузера, работать на устройстве не может.

Устройства, в рамках данного исследования, можно разделить на мобильные – смартфоны, планшеты и прочие устройства, зачастую использующие исключительно виртуальные способы ввода, и настольные – стационарные компьютеры и ноутбуки – то есть устройства, зачастую использующие, преимущественно, физические устройства ввода. Различия в устройствах ввода позволяют классифицировать устройства, но решающую роль в особенностях взаимодействия устройства пользователя и веб-страницы играет конфигурация оборудования, которым оснащено устройство.

В случае с компьютерами, зачастую, имеется постоянное подключение к интернету, нередко, по проводам. Одновременно с постоянным подключением к сети, в распоряжении браузера, запущенного на компьютере, зачастую имеется также большая вычислительная мощность устройства (достаточно большая для запуска практически всех веб-приложений на практически любом компьютере, исключения представляют из себя игры с высокими техническими требованиями и специализированное программное обеспечение (далее – ПО), осуществляющее объёмные вычисления, например, ПО для расчёта аэродинамики), а также большой объём оперативной памяти, что также позволяет веб-странице производить сложные и объёмные вычисления.

Но, когда браузер запущен на мобильном устройстве, то в его распоряжении, зачастую, оказываются небольшие, по сравнению с

компьютерами, вычислительные мощности и объём оперативной памяти. Следствием из данного обстоятельства является то, что на мобильном устройстве работа многих веб-страниц невозможна в полноценном режиме, и пользователь, при работе с данной веб-страницей, с большой вероятностью столкнётся с низким количеством кадров отрисовки веб-страницы или даже с её полной неработоспособностью.

Эти особенности работы веб-страниц, в частности тех, веб-страниц, которые используют фреймворки, то есть дополнительные “слои”, которые позволяют быстрее разрабатывать качественные веб-приложения, зачастую, с потерей в их быстродействии, вызваны большим количеством “посредников”, которые интерпретируют код предыдущего уровня для кода следующего уровня.

При этом, при использовании браузера на компьютере, это количество программных посредников позволяет браузеру не выходить за рамки имеющейся вычислительной мощности и оперативной памяти, из-за чего пользователь имеет возможность полноценно использовать все возможности интерактивных веб-страниц.

#### 1.1.2. Исследование популярных устройств и их характеристик

Для анализа текущей ситуации, в которой оказывается веб-страница, запущенная на устройстве, необходимо проанализировать текущую ситуацию с самими устройствами, что <sup>2</sup>позволит сделать вывод о том, какими суммарными ресурсами располагают программы, запущенные на смартфоне, ноутбуке или ином устройстве. В первую очередь, необходимо проанализировать, какие модели соответствующих типов устройств стали самыми популярными в одинаковый недавний период, что позволит определить, по модели устройства его техническую конфигурацию.

В 2020 году, по данным Группы “М.Видео-Эльдорадо” год самыми продаваемыми моделями ноутбуков стали[1]:

- Huawei MateBook D 14 Nbl-WAQ9R;

- Huawei MateBook D 15 Boh-WAQ9R;
- Honor MagicBook 14 Nbl-WAQ9HNR;
- Lenovo IdeaPad S145-15AST;
- Apple MacBook Air 13 (MQD32);

Данные ноутбуки имеют следующие процессоры (таблица 1).

Таблица 1. Соотношение моделей процессоров и самых популярных моделей ноутбуков в 2020 году

Ноутбук	Модель процессора
Huawei MateBook D 14 Nbl-WAQ9R	AMD Ryzen 5 3500U[2]
Huawei MateBook D 15 Boh-WAQ9R	AMD Ryzen 5 3500U[3]
Honor MagicBook 14 Nbl-WAQ9HNR	AMD Ryzen 5 3500U[4]
Lenovo IdeaPad S145-15AST	AMD A6 9225[5]
Apple MacBook Air 13 (MQD32)	Core i5 5350U[6]

В таблице 1 видно, какие модели процессоров устанавливались в самые распространённые ноутбуки, поэтому следующим шагом было определение средних характеристик данных процессоров (таблица 2).

Таблица 2. Основные характеристики самых распространённых процессоров ноутбуков в 2020 году

Наименование	Частота, ГГц	К-во ядер	К-во потоков
AMD Ryzen 5 3500U	2,1[7]	4[7]	8[7]
AMD A6 9225	2,6[8]	2[8]	2[8]
Core i5 5350U	1,8[9]	2[9]	4[9]
<i>Среднее значение</i>	2,17	2,67	4,67

Как видно из таблицы 2, в среднем, ноутбуки располагают следующими вычислительными характеристиками:

- Частотой процессора – 2,17 ГГц;
- Количеством ядер процессора – 2,67;

- Количеством потоков процессора – 4,67;

Далее было необходимо определить средние объёмы оперативной и постоянной памяти самых распространённых ноутбуков (таблица 3).

Таблица 3. Объёмы оперативной памяти у самых популярных ноутбуков в 2020 году

Ноутбук	Объём оперативной памяти, ГБ	Объём постоянной памяти, ГБ
Huawei MateBook D 14 Nbl-WAQ9R	8 [2]	512 [2]
Huawei MateBook D 15 Boh-WAQ9R	8 [3]	256 [3]
Honor MagicBook 14 Nbl-WAQ9HNR	8 [4]	256 [4]
Lenovo IdeaPad S145-15AST	8 [5]	256 [5]
Apple MacBook Air 13 (MQD32)	8 [6]	128 [6]
<i>Среднее значение</i>	8	281,6

Как видно из таблицы 3, самые популярные ноутбуки 2020 года имели по 8 ГБ оперативной памяти и, в среднем, по 281,1 ГБ постоянной памяти.

Следующим шагом, было необходимо провести аналогичное исследование, связанное с самыми популярными смартфонами.

По данным Группы “М.Видео-Эльдорадо”, в 2020 году самыми популярными моделями смартфонов были[10] следующие:

- Samsung Galaxy A51 (64 ГБ);
- Samsung Galaxy A10;
- Samsung Galaxy A01;
- iPhone 11 (128 ГБ);
- Samsung Galaxy A51 (128 ГБ);

Соответственно, данные смартфоны имели процессоры со следующими характеристиками (таблица 4).

Таблица 4. Соотношение моделей процессоров и самых продаваемых моделей смартфонов в 2020 году

Смартфон	Модель процессора
Samsung Galaxy A51 (64 ГБ)	Samsung Exynos 9611 [11]
Samsung Galaxy A10	Samsung Exynos 7884B [12]
Samsung Galaxy A01	Qualcomm Snapdragon 439 [13]
iPhone 11 (128 ГБ)	Apple A13 Bionic [14]
Samsung Galaxy A51 (128 ГБ)	Samsung Exynos 9611 [15]

В таблице 4 приведены процессоры, которые устанавливались в самые продаваемые модели смартфонов в 2020 году.

Далее были определены их средние характеристики (таблица 5)

Таблица 5. Основные характеристики процессоров самых продаваемых смартфонов 2020 года

Наименование	Частота, ГГц	К-во ядер	К-во потоков
Samsung Exynos 9611	2,3x4 + 1,7x4 [16]	4+4 [16]	8 [17]
Samsung Exynos 7884B	1,6x2+1,35x6[18]	2+6 [18]	8 [18]
Qualcomm Snapdragon 439	1,95x4+1,45x4[19]	4+4 [19]	8 [19]
Apple A13 Bionic	2,65x2+1,8x4[20]	2+4 [20]	6 [21]
Samsung Exynos 9611	2,3x4+1,7x4 [22]	4+4 [22]	8 [23]
<i>Среднее значение</i>	1,83	7,6	7,6

Как видно из таблицы 5, в среднем, смартфоны в 2020 году обладают следующими характеристиками:

- Частотой процессора – 1,83 ГГц;
- Количеством ядер процессора – 7,6;
- Количеством потоков процессора – 7,6;

Далее было необходимо определить средние объёмы оперативной и постоянной памяти самых продаваемых смартфонов (таблица 6).

Таблица 6. Объёмы оперативной памяти у самых продаваемых смартфонов в 2020 году

Смартфон	Объём оперативной памяти, ГБ	Объём постоянной памяти, ГБ
Samsung Galaxy A51 (64 ГБ)	4 [11]	64 [11]
Samsung Galaxy A10	2 [12]	32 [12]
Samsung Galaxy A01	2 [13]	16 [13]
iPhone 11 (128 ГБ)	4 [24]	128 [14]
Samsung Galaxy A51 (128 ГБ)	6 [15]	128 [15]
<i>Среднее значение</i>	3,6	73,6

Как видно из таблицы 6, в среднем, самые продаваемые в 2020 году смартфоны имели 3,6 ГБ оперативной памяти и 73,6 ГБ постоянной памяти.

Далее, было необходимо сравнить данные характеристики смартфонов и ноутбуков (таблица 7).

Таблица 7. Сравнение основных характеристик смартфонов и ноутбуков

Характеристика	Смартфоны	Ноутбуки
Частота процессора, ГГц	1,83	2,17
К-во ядер процессора, шт	7,6	2,67
К-во потоков процессора, шт	7,6	4,67
Объём оперативной памяти, ГБ	3,6	8
Объём постоянной памяти, ГБ	73,6	281,6

В соответствии с данными из таблицы 7, смартфоны, по сравнению с ноутбуками, имеют:

- На 15,6% меньшую частоту процессора
- В 2,85 раз большее количество ядер процессора

- В 1,63 раза большее количество потоков процессора
- В 2,23 раза меньший объём оперативной памяти
- В 3,83 раза меньший объём постоянной памяти

В соответствии с результатами сравнения, можно сделать вывод, что суммарная максимальная производительность всех ядер (произведение количества ядер на частоту процессора) самых распространенных смартфонов превышает аналогичную характеристику ноутбуков в 2,4 раза, также как и среднее количество потоков у смартфонов больше на 63%, но эта характеристика только позволяет смартфону в целом лучше справляться с большим количеством выполняющихся одновременно приложений, но не обеспечивать лучшие условия выполнения каждому отдельно взятому приложению.

В то же время меньший в 2,23 раза объём оперативной памяти ставит любое запущенное на смартфоне приложение в значительно худшие условия, чем на ноутбуках.

Безусловно, приложения могут быть оптимизированными и многопоточными, но веб-страница, которая должна полностью функционировать на экране и за его пределами, нуждается в большом количестве оперативной памяти и далеко не всегда может быть эффективно выполнена во многих потоках параллельно, из-за чего можно считать частоту процессора и объём оперативной памяти более важными характеристиками для работы веб-приложений, чем количество ядер и количество потоков процессора.

Также, меньший в 3,83 раза объём постоянной памяти мобильных устройств не позволяют приложению как иметь большой объём самого приложения, так и хранить файлы большого размера (например, объёмом более 16 ГБ), так как минимальное значение постоянной памяти среди самых популярных смартфонов составило 16 ГБ.

Вышеприведённые обстоятельства являются основными предпосылками, формирующими главные тенденции во взаимодействии веб-страниц с устройством пользователя в браузерах.

## 1.2. Исследование основных тенденций во взаимодействии веб-страниц с устройством пользователя

### 1.2.1. Исследование тенденций в предоставляемом пользователям ПО среди популярных сервисов

Так как веб-страницы, которые выполняются на мобильных устройствах, вынуждены работать в условиях малых объёмов оперативной памяти и небольшой вычислительной мощности, это затрудняет создание веб-приложений, выполняющих какие-либо ресурсозатратные функции. Вследствие этого, необходимо было изучить, какие программные продукты предоставляют своим пользователям наиболее популярные сервисы.

Для анализа ситуации и текущих тенденций, необходимо было проанализировать, какой набор ПО предоставляют своим пользователям самые популярные онлайн-сервисы России на текущий момент, и какой набор ПО они предоставляли ранее.

Для начала, необходимо было исследовать, какие интернет-сайты являются были самыми популярными в России в 2020 году.

По данным Alexa Internet, самыми посещаемыми сайтами российского сегмента интернета в 2020 стали сайты следующих сервисов [25]:

- YouTube;
- ВКонтакте;
- Google (группа сервисов);
- Яндекс (группа сервисов);
- Mail.ru (группа сервисов);
- Авито;
- Одноклассники;



- Wikipedia;
- AliExpress;
- Сбербанк Онлайн;

Далее, было необходимо было проанализировать, какие мобильные приложения были самыми востребованными в 2020 году в России. По данным App Annie, в 2020 году самыми востребованными приложениями стали приложения следующих сервисов[26]:

- TikTok;
- WhatsApp;
- Telegram;
- Сбербанк Онлайн;
- Instagram;
- Likee;
- Госуслуги;
- Авито;
- ВКонтакте;
- YouTube;
- AliExpress;
- ЛитРес;
- Одноклассники;

После анализа двух данных списков, было выявлено, что у многих сервисов, сайты которых являются самыми посещаемыми в России, также имеются мобильные приложения, при этом, эти приложения являются одними из самых востребованных среди пользователей (таблица 8).

Таблица 8. Наличие у самых популярный онлайн-сервисов России сайтов-лидеров и приложений-лидеров

Сервис	Сайт-лидер	Приложение-лидер
TikTok	Нет	Да
WhatsApp	Нет	Да

Telegram	Нет	Да
Сбербанк Онлайн	Да	Да
Instagram	Нет	Да
Likee	Нет	Да
Госуслуги	Нет	Да
Авито	Да	Да
ВКонтакте	Да	Да
YouTube	Да	Да
AliExpress	Да	Да
ЛитРес	Нет	Да
Одноклассники	Да	Нет
Google	Да	Нет
Mail.ru	Да	Нет
Wikipedia	Да	Нет
Яндекс	Да	Нет

Как видно из таблицы 8, у многих сервисов, у которых есть сайт-лидер (лидер, в данном случае, означает, что данное ПО стало одним из самых востребованных в России в рассматриваемый промежуток времени), есть также приложение, которое также является одним из самых востребованных.

В том случае, когда у некоторого сервиса есть сайт-лидер, но мобильное приложение отсутствует или не является одним из самых востребованных, можно сделать вывод, что клиентское веб-приложение не требует значительных ресурсов (как, например, сервис Wikipedia, так как на интернет-страницах данного ресурса, в основном, размещена только текстовая информация, а сами страницы не являются интерактивными, из-за чего количество необходимой оперативной памяти для удобного функционирования данных страниц достаточно невелико), либо сложные для выполнения в браузере сервисы были

выделены в отдельные приложения для мобильных устройств, как в случае с группами онлайн-сервисами, объединёнными в Mail.ru и Яндекс.

В том же случае, когда у сервиса есть и одно из самых популярных приложений, и один из самых популярных интернет-сайтов, можно сделать вывод, что выполнение веб-приложений, требующих значительных ресурсов для удобства использования, на мобильном устройстве невозможно или затруднительно, в то время как открытие данных сайтов в браузере на компьютере будет удобно.

Далее, была проанализирована динамика развития перечня предлагаемого пользователям ПО онлайн-сервисов, имеющих, на данный момент, как один из самых популярных сайтов, так и одно из самых востребованных приложений (таблица 9).

Таблица 9. Сравнение времени появления продуктов-лидеров рынка у самых популярных сервисов России

Сервис	Запуск сайта-лидера	Запуск приложения-лидера	Появилось раньше
Сбербанк Онлайн	2008 год [27]	2012 год [28]	Сайт
Авито	2007 год [29]	2011 год [30]	Сайт
ВКонтакте	2006 год [31]	2011 год [32]	Сайт
YouTube	2005 год [33]	2010 год [34]	Сайт
AliExpress	2010 год [35]	2012 год [36]	Сайт

Как видно из таблицы 9, существует тенденция к адаптации сервисов под устройства пользователей: изначально, как более универсальный вариант, сервисы предоставляют пользователям веб-приложение, как единственный вариант доступа к использованию сервиса, но, так как на мобильных устройствах использование сайтов с большими функциональными возможностями, зачастую, затруднено, то выпускается мобильное приложение.

Это позволяет пользователям использовать сервис на как на компьютерах – с помощью сайта, так и на мобильных устройствах – с помощью приложения.

Таким образом, можно сделать вывод, что, для использования сервиса на компьютере, наиболее востребованным видом ПО является веб-приложение, размещённое на веб-странице.

### 1.2.2. Исследование основных тенденций, связанных с полномочиями веб-страниц в популярных браузерах

Далее, было произведено исследование, как изменялась ситуация, связанная с полномочиями веб-страницы, запущенной на компьютере. Для этого, прежде всего, необходимо было определить, какие браузеры, на данный момент, являются самыми популярными. По данным Statcounter [37] на февраль 2021 года, самыми популярными браузерами являются:

- Chrome (год выхода – 2008 [38]);
- Safari (год выхода – 2006 [39]);
- Firefox (год выхода – 2002 [40]);

Далее было необходимо рассмотреть основные тенденции, связанные с полномочиями веб-страниц. Прецедентом изменения полномочий веб-страниц было решено считать упоминание объекта взаимодействия в описании отличий определённой версии браузера от предыдущей версии (таблица 10).

Таблица 10. Хронология расширения полномочий веб-страниц

Объект взаимодействия	Год появления в Chrome	Год появления в Safari	Год появления в Firefox
Геолокация	2010 [41]	2010 [42]	2009 [43]
Уведомления	2010 [44]	2013 [44]	2011 [44]
Веб-камера и микрофон	2012 [45]	2017 [46]	2013 [47]

Как видно из данных, которые приведены в таблице 10, количество функций устройства, с которыми может взаимодействовать веб-страница, постепенно увеличивается во всех самых популярных браузерах. Из данных сведений можно сделать вывод, что во всех самых популярных браузерах существует следующая тенденция: браузеры постепенно увеличивают полномочия веб-страниц. Так как это свойственно самым популярным браузерам, можно сделать вывод, что данная тенденция весьма важна, является актуальной, и, вероятно, будет продолжаться.

Одно из важных полномочий, которое есть у программ, которые запускаются в операционной системе (далее – ОС) [48, с. 74], но которого нет у веб-страниц, так как они выполняются в браузере, а браузер не предоставляет им данных полномочий, это работа с файлами. Безусловно, у веб-страницы есть возможность сохранять данные на устройстве пользователя, в частности, в виде cookies и локального хранилища, но эти данные будут доступны только внутри браузера, и будут доступны только страницам одного домена.

Несмотря на то, что данная особенность взаимодействия устройства пользователя с веб-страницами весьма полезна для защиты от действий вредоносных веб-страниц, так как они лишены возможности работать с файлами пользователя [49, с. 47], у данного подхода, практикуемого браузерами, имеется существенный недостаток: это не позволяет пользователю использовать внутри браузера, для работы с файлами устройства, многие виды программного обеспечения без скачивания, например:

- Редакторы текста и программного кода, позволяющие использовать автоматизацию для более эффективной разработки [50, с. 459];
- Графические редакторы;
- Архиваторы;
- Программы просмотра файлов специализированного типа – то есть файлов, которые открываются в непопулярном, нераспространённом или

профессиональном ПО, в том числе, в ПО, для работы с которым необходима действующая лицензия;

Безусловно, браузер позволяет просматривать изображения различных популярных форматов, а также просматривать многие типы файлов в текстовом виде, так как текст и изображения являются основными частями веб-страниц, а потому могут быть открыты отдельно от веб-страницы.

Также, если работа с файлами определённого типа требует больших вычислительных ресурсов, как, например, работа с видеофайлами или работа с трёхмерными объектами, то работа с ними в браузере нецелесообразна, так как браузер только увеличит суммарные затраты ресурсов устройства на работу с данными файлами.

Работа с файлами в браузере также имеет малую целесообразность на мобильных устройствах, так как для условия, с точки зрения вычислительных ресурсов и памяти, в которых выполняется веб-страница на мобильных устройствах, значительно хуже, чем на компьютерах.

Безусловно, если работа с файлами осуществляется с веб-страницы, необходимо проверять надёжность данной веб-страницы: данный принцип используется в TLS<sup>1</sup>, при этом браузер проверяет надёжность сайта с помощью проверки сертификата безопасности, предоставляемого сайтом.

Из вышеперечисленного можно сделать следующий вывод: разработка ПО для расширения полномочий доверенных страниц позволит решить актуальную проблему отсутствия у веб-страниц полномочий для работы с файлами пользователя на компьютере.

---

<sup>1</sup> Документация TLS [Интернет-ресурс]  
URL: <https://datatracker.ietf.org/wg/tls/documents/> (дата обращения: 06.05.2021)

## ГЛАВА 2. ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ РАСШИРЕНИЯ ПОЛНОМОЧИЙ ДОВЕРЕННЫХ ВЕБ-СТРАНИЦ

### 2.1. Исследование возможности и способов расширения полномочий веб-страниц

#### 2.1.1. Исследование алгоритма взаимодействия веб-страницы и пользователя

Для исследования возможности и способов расширения полномочий веб-страниц, необходимо, прежде всего, рассмотреть, по какому алгоритму работают веб-страницы (рисунок 1).

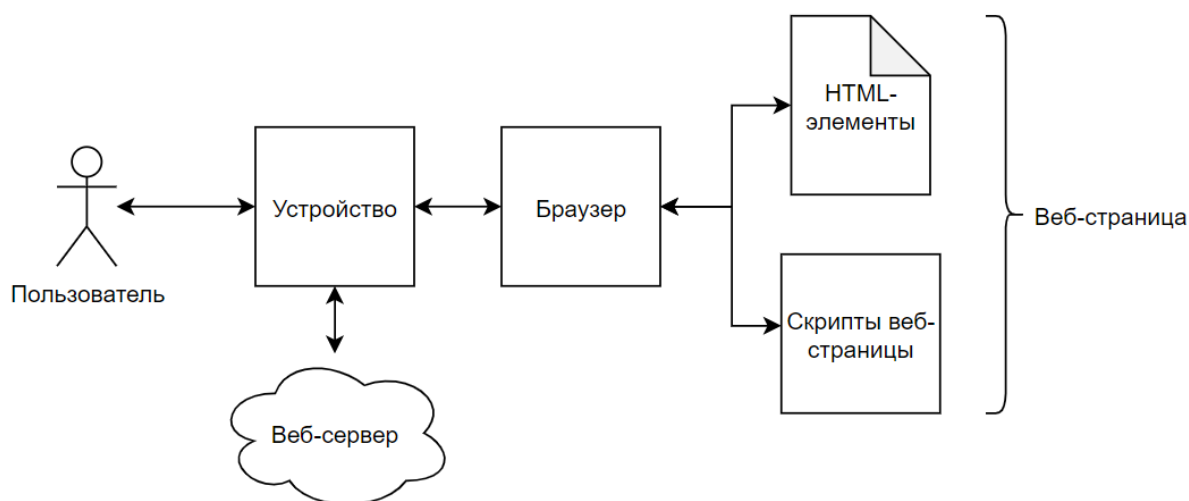


Рисунок 1. – Взаимодействие веб-страницы и устройства пользователя

Как видно из схемы на рисунке 1, веб-страница, которая, по умолчанию, загружается в браузер с веб-сервера (хоть и может быть загружена из локального файлового хранилища устройства) представляет из себя совокупность следующих компонентов:

- HTML-элементов, из которых состоит видимая пользователю часть веб-страницы. Данный компонент крайне важен по причине того, что как графический интерфейс любого веб-приложения состоит из данных элементов, хотя веб-приложение может и не иметь графического интерфейса. Технология, которая используется браузером для создания

интерфейса веб-страницы[51, с. 12] из элементов – HTML (от англ. HyperText Markup Language – «язык разметки гипертекста»). Веб-страница может содержать в себе только HTML-элементы и не иметь какого-либо интерактивного контента. В случае, если веб-страница не интерактивна, для того, чтобы изменить отображаемый на экране контент, пользователю потребуется перейти на другую веб-страницу;

- Скриптов веб-страницы, представляющих из себя совокупность приложений на следующих технологиях:
  - JavaScript – технология, которая позволяет создавать интерактивные веб-страницы, добавляя в них код программ на языке JavaScript. Данная технология позволяет запускать в рамках веб-страницы приложения, и выполнять программы, запускаемые, в том числе, при взаимодействии пользователя с HTML-элементами на странице. Данная технология позволяет веб-странице использовать различные функции браузера, в том числе, взаимодействовать, например, с устройствами ввода и вывода, а также сохранять на устройстве пользователя переменные;
  - CSS – технология, позволяющая определять внешний вид элементов. Данная технология крайне важна, так как от её работы зависит, зачастую, не только внешний вид веб-страницы, но и само взаимодействие пользователя с её функционалом. Таким образом, несмотря на то, что данная технология напрямую не может взаимодействовать с функционалом браузера, от неё, во многом, зависит взаимодействие веб-страницы с пользователем;

Также, как видно из схемы на рисунке 1, веб-страница, фактически, отделена от устройства пользователя браузером. Браузер, безусловно, предоставляет веб-странице минимально необходимые полномочия, которых достаточно для, например, отображения на экране текста и изображений, но которых не хватает для, например, управления устройством пользователя.



Данная политика, безусловно, обусловлена стремлением разработчиков браузеров обезопасить пользователя от небезопасного содержимого веб-страниц, но, в том же время, она значительно ограничивает возможности веб-страниц.

Веб-страницы, имеющие полномочия, близкие к приложениям, запущенным в операционной системе, позволили бы пользователям, во многих случаях, не скачивать приложения, а значит всегда иметь, в том числе, последнюю версию приложений, что положительно скажется на удобстве и безопасности использования данных приложений. Вследствие этого, необходимо рассмотреть, каким образом можно расширить полномочия веб-страниц.

Непосредственно веб-страница может иметь расширенные полномочия только при условии, что браузер, в котором веб-страница выполняется, предоставит их ей. Несмотря на то, что многие браузеры имеют возможность установки расширений, которые расширяют полномочия веб-страниц, в ведущих браузерах нет таких полномочий расширений, как, например, работа с файловой системой устройства пользователя.

#### 2.1.2. Исследование возможности расширения полномочий веб-страниц с помощью специализированного браузера

Это означает, что для выполнения поставленной задачи, в том случае, если расширение полномочий доверенных веб-страниц будет осуществлено с помощью браузера, необходимо создание принципиально нового, специализированного браузера. В случае использования данного браузера, взаимодействие веб-страниц с устройством возможно.

Разумеется, взаимодействие непосредственно с устройством можно не рассматривать, так как необходимо рассматривать тот уровень приложений, на котором функционируют браузеры, из-за чего целесообразно рассматривать взаимодействие веб-страницы, через специализированный браузер, с ОС,

которая, в свою очередь и имеет возможность взаимодействовать с устройством, а, следовательно, и с пользователем.

Таким образом, взаимодействие веб-страницы с ОС может быть следующим (рисунок 2).

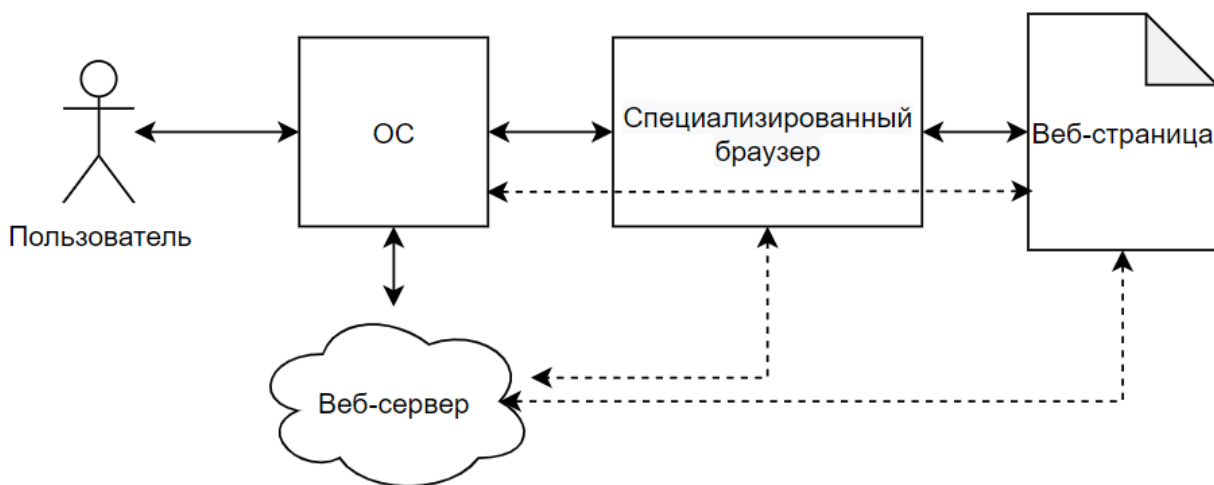


Рисунок 2. – Взаимодействие веб-страницы с устройством через специализированный браузер

Как показано на рисунке 2 сплошными линиями, непосредственно с ОС может взаимодействовать специализированный браузер, а веб-страница, в таком случае, сможет взаимодействовать с ОС и веб-сервером через данную программу, что обозначено на рисунке 2 пунктирными линиями.

Такой, специальный, браузер, безусловно, может быть создан, но, в сущности, он будет представлять из себя обычный браузер, так как важно сохранить весь существующий функционал веб-страниц, только лишь с тем отличием, что у веб-страниц появятся все полномочия приложений ОС. Вместе с этим, крайне важно использовать наиболее современные и безопасные подходы, касающиеся взаимодействия браузера с обычными веб-страницами. Это значительно усложнит разработку данного ПО, так как, помимо выполнения основной задачи – расширения полномочий доверенных веб-страниц, будет необходимо также решать те же задачи, что и всем браузерам. В частности, появится необходимость весьма частых обновлений функционала и

безопасности, чтобы по своему функционалу браузера и особенностям данный браузер не был хуже наиболее популярных браузеров.

### 2.1.3. Исследование возможности расширения полномочий веб-страниц с помощью дополнительного ПО и браузера

Альтернативой созданию специализированного браузера является разработка дополнительного ПО, которое позволит веб-странице иметь расширенные полномочия, а пользователю – пользоваться любым, в том числе самым популярным, браузером (рисунок 3).

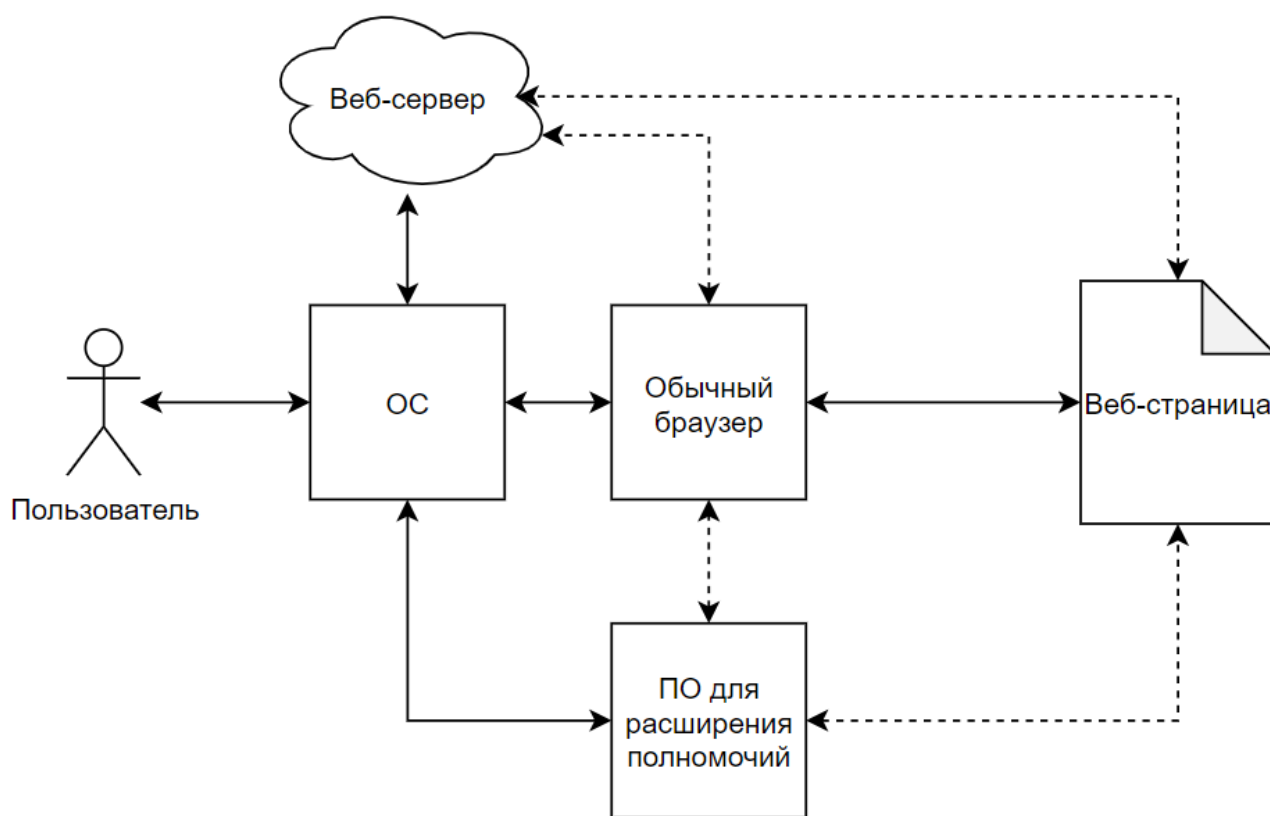


Рисунок 3. – Схема взаимодействия веб-страницы и ПО для расширения полномочий

Как видно из схемы на рисунке 3, доверенная веб-страница будет взаимодействовать с устройством пользователя через ПО для расширения полномочий независимо от браузера. Это положительно скажется на безопасности и продуктивности взаимодействия пользователя как с обычными

веб-страницами, так и с доверенными веб-страницами, имеющими расширенные полномочия.

Безусловно, напрямую веб-страница в обычном браузере взаимодействовать с каким-либо ПО не имеет возможности, поэтому, как обозначено на схеме рисунке 3 пунктирными линиями, взаимодействие возможно, но не прямое, а через посредника – браузер.

Исходя из результатов проведённого исследования, было решено выполнить поставленную задачу без создания специализированного браузера, но с сохранением вышеописанной концепции: расширение полномочий доверенных веб-страниц будет осуществляться с помощью двух приложений, одновременно запущенных на устройстве пользователя:

- Обычного браузера, так как это позволит пользователю использовать привычный интерфейс взаимодействия с веб-страницами, использовать преимущества ведущих браузеров, при том, что пользователь сможет пользоваться веб-страницами с расширенными полномочиями, самостоятельно выбирая наиболее удобный браузер;
- Приложения, которое будет предоставлять веб-страницам, запущенным в браузере, расширенные полномочия, обмениваясь с данными веб-страницами информацией через локальную сеть устройства;

Данная архитектура позволяет выполнить поставленную задачу за минимальный срок, с минимальными трудозатратами, и, одновременно, с наибольшей продуктивностью работы пользователя, так как, например, смена браузера негативно бы отразилась на комфорте пользователя, в частности, из-за смены интерфейса браузера.

Также, данная архитектура является популярной, так как активно используется во многих веб-приложениях: многие веб-приложения, хотя также выполняются на устройстве пользователя, в браузере, но непосредственную работу с файлами, расположенную на сервере, осуществляют с помощью

коммуникации с приложением, выполняющимся на сервере данного приложения (рисунок 4).

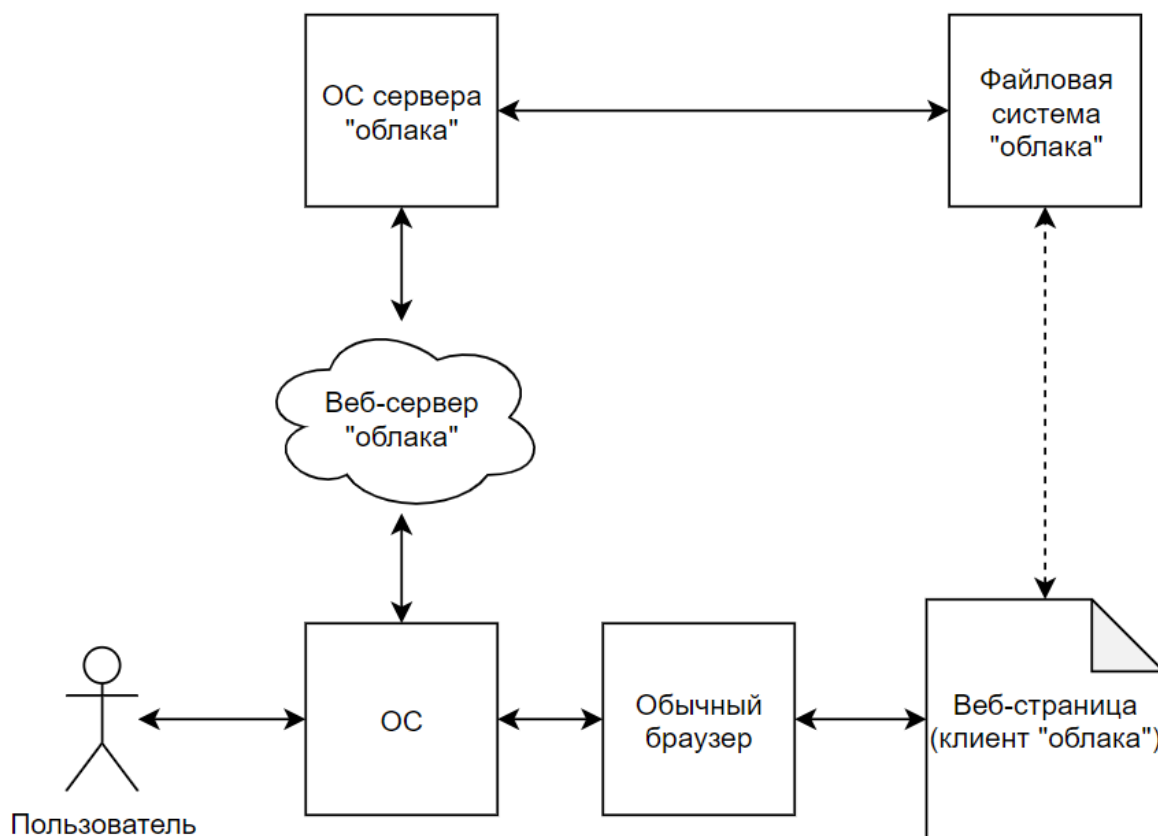


Рисунок 4. – Общая схема взаимодействия “облачного” веб-приложения

Как показано на схеме на рисунке 4, при использовании, например, какого-либо облачного хранилища файлов, веб-страница, которая, в данном случае, является клиентом веб-приложения (и, как показано сплошными линиями, фактически может взаимодействовать только с браузером) позволяет выполнять операции над файловой системой удалённого устройства (что показано на схеме пунктирными линиями), отправляя команды серверу данного “облачного” веб-приложения и отображая результат их выполнения, например, в графическом интерфейсе веб-страницы, состоящем из HTML-элементов.

Фактически, при использовании выбранной в результате проведённого исследования архитектуры, устройства, на которых будут работать обычный браузер, в котором будет выполняться доверенная страница с расширенными полномочиями, то есть клиент веб-приложения, и работать веб-сервер,

работающий с файловой системой “облака”, совпадут, из-за чего взаимодействие всех компонентов будет организовано следующим образом (рисунок 5).

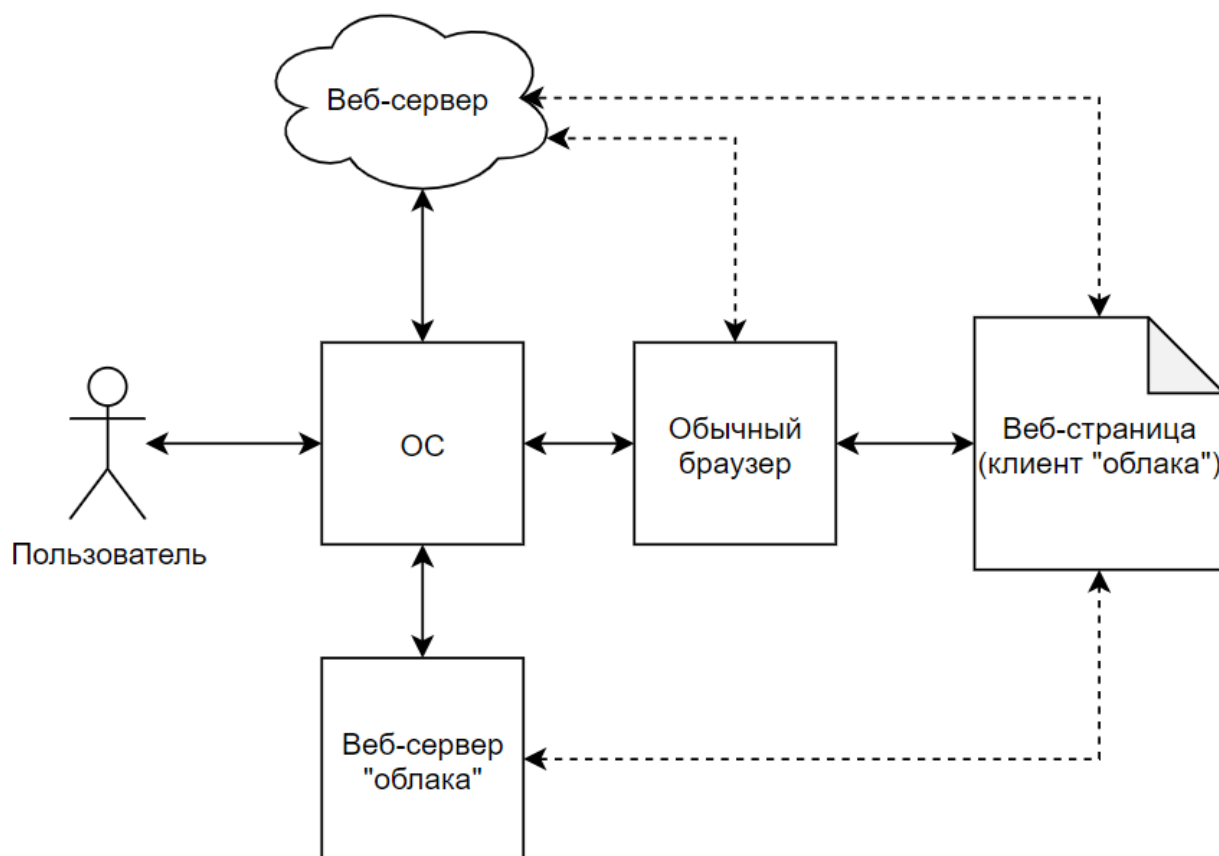


Рисунок 5. – Схема взаимодействия веб-страницы с локальным “облачным” приложением

Как показано на схеме на рисунке 5, у веб-страницы может быть свой веб-сервер, который будет отвечать за её формирование, хотя она может быть загружена и из локального хранилища. С веб-сервером у веб-страницы может быть какое-либо взаимодействие через посредников, как показано на схеме пунктирными линиями, также как и с локальным “облачным” веб-сервером, в то время как непосредственно с ОС будут взаимодействовать обычный браузер и веб-сервер “облака”, также как и веб-сервер веб-страницы (через сетевое подключение).

Другими словами, данная архитектура позволит сделать “облачным” локальное веб-приложение: непосредственно веб-приложение будет

осуществлять лишь взаимодействие с пользователем через, например, графический интерфейс и отправлять команды на работу с файлами “серверной” части архитектуры, размещаемой также на компьютере пользователя, которая, в свою очередь, будет уже непосредственно взаимодействовать с файловой системой устройства пользователя.

Данная архитектура позволяет считать веб-страницу, в совокупности с ПО для расширения её полномочий, веб-приложением, так как имеется часть приложения, отвечающая интерфейс, и часть приложения, отвечающая за работу с файлами [52, с.26].

2.1.4. Определение списка необходимых операций, выполняемых дополнительными ПО для расширения полномочий веб-страниц

Таким образом, основной функцией компонента веб-приложения, отвечающего за взаимодействие с “серверной” частью, будет являться отправка команд для взаимодействия с файловой системой пользователя и получение результата выполнения отправленных команд.

Для того, чтобы веб-приложение могло определить результат выполнения отправленной команды, все команды должны возвращать код выполнения запрашиваемой операции, который может принимать следующие значения:

- Успешное выполнение команды
- Взаимодействие с файлами не было осуществлено успешно (например, отсутствует запрашиваемый файл, или невозможно удалить файл из-за того, что текущий пользователь не имеет прав на удаление файла)

Так как основных команд, связанных с работой с файловой системой, всего несколько, их и необходимо реализовать, так как многие действия, которые выполняет пользователь над файловой системой, являются комбинацией базовых действий, например:

- При изменении содержимого файла, фактически, над файловой системой выполняются две последовательные операции (в том числе, со значительной разницей во времени):
  - Чтение файла;
  - Запись файла (с новыми данными, но с тем же наименованием и в том же расположении);
- При копировании файла, особенно, если не рассматривать функционирование буфера обмена, происходят, зачастую, одновременно, две операции:
  - Чтение файла;
  - Запись файла (с тем же содержимым, но с отличным наименованием и/или расположением);
- Перемещение файла можно рассмотреть также как три независимые операции, осуществляемые последовательно:
  - Чтение файла;
  - Запись файла (в новое расположение и/или с новым наименованием, аналогично операции копирования);
  - Удаление файла (который был прочитан);

Массовые операции, то есть, например, копирование папки с содержимым или же переименование некоторого списка файлов, можно рассматривать как последовательное выполнение нескольких команд.

Также, безусловно, существуют операции, которые нельзя заменить комбинацией базовых команд, например, следующие:

- Получение списка файлов в папке;
- Получение списка имеющихся в компьютере дисков, подключённых к файловой системе (данная операция необходима для, например, работы с жёсткими дисками или картами памяти);
- Создание папки;



Таким образом, с учётом возможности применения комбинаций и последовательностей в использовании команд, был определён список команд, которыми будет оперировать веб-приложение с расширенными с помощью вышеописанного ПО полномочиями:

- “Список” – получение списка файлов и папок:
  - Принимает в качестве аргумента путь папки;
  - Возвращает код выполнения команды и два списка: список файлов в запрошенной папке и список папок в ней, а также основную информацию о каждом файле:
    - Наименование файла или папки;
    - Абсолютный путь файла или папки;
    - Размер (только для файлов);
    - Время создания файла или папки;
    - Время изменения файла или папки;
- “Чтение” – получение содержимого файла по указанному пути файла:
  - Принимает в качестве аргумента путь файла, содержимое которого нужно прочитать;
  - Возвращает код выполнения команды и содержимое файла в виде списка байтов, что необходимо для работы с любыми типами файлов, в том числе, с изображениями, архивами и т.п. типами файлов;
- “Запись” – запись данных в файл, расположенный по определённому пути:
  - Принимает в качестве аргументов путь записываемого файла (как существующего, так и ранее не существовавшего в файловой системе) и данные, которые требуется в него записать в виде списка байтов;
  - Возвращает код выполнения команды;
- “Удаление” – удаление файла или папки по определённому пути;

- Принимает в качестве аргумента путь удаляемого файла или папки;
- Возвращает код выполнения операции;
- “Создание папки” – создание новой папки по определённому пути:
  - Принимает в качестве аргумента путь папки, которую необходимо создать;
  - Возвращает код выполнения операции;
- “Переименование” – изменение имени файла файла по определённому пути. <sup>2</sup> Несмотря на то, что данная команда может быть выполнена с помощью выполнения комбинации из команд чтения удаления и записи файла (или создания папки), но, в случае с папками, данная операция потребует выполнения рекурсивного перемещения, что нецелесообразно для такой примитивной операции:
  - Принимает в качестве аргументов путь файла или папки, которую необходимо переименовать, а также новое наименование файла или папки соответственно;
  - Возвращает код выполнения операции;

#### 2.1.5. Составление алгоритма работы ПО для расширения полномочий веб-страниц

Таким образом, оптимальный порядок действий для выполнения команды, расширяющей полномочия веб-страницы, следующий (рисунок 6).

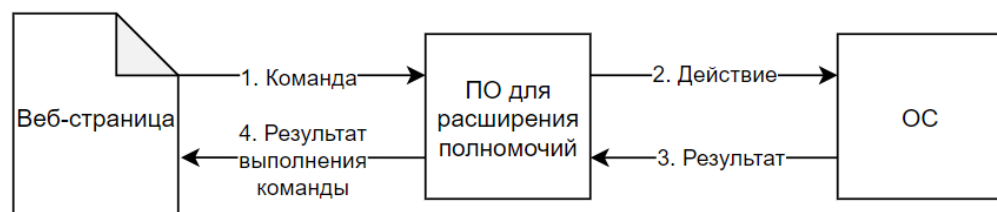


Рисунок 6. – Общий порядок действий при выполнении команды

Как продемонстрировано на схеме на рисунке 6, выполнение команды должно производиться в 4 следующих этапа:

1. Веб-страница формирует и отправляет команду приложению, которое её выполнит. Данное действие, например, может быть вызвано событием взаимодействия пользователя и интерфейсом веб-страницы. Коммуникация, безусловно, осуществляется через посредников, в частности, через браузер и сетевые протоколы внутри ОС, но благодаря стандартизации, существующей, в том числе, у браузеров, можно рассматривать, в рамках функционирования ПО для расширения полномочий, что коммуникация осуществляется напрямую между веб-страницей и данным ПО;
2. Команда от веб-приложения принимается и выполняется ПО для расширения полномочий с помощью обращения к стандартным средствам ОС для работы с устройством. Это взаимодействие происходит напрямую по той причине, что ПО для расширения полномочий является приложением операционной системы. Соответственно, в качестве аргументов, которые требуются ОС для выполнения необходимого веб-странице действия, используются те аргументы, которые были получены от веб-страницы;
3. ПО для расширения полномочий получает от ОС результат выполнения команды. Из данных, которые получает данное ПО от ОС по завершению выполнения команды, формируется ответ веб-страницы. В рамках данного действия возможна ситуация, когда успешно завершить выполнение запрашиваемого действия у ОС не было возможности (например, по причине некорректных аргументов). Неуспешное завершение выполнения запрашиваемого веб-страницей действия является штатной ситуацией, которая обрабатывается ПО для расширения полномочий и включается в результат выполняемой команды;
4. Веб-страница получает результат выполнения команды, сформированный ПО для расширения полномочий. Формат результата, который получает

веб-страница, зависит от того, какая команда была инициирована, а также от того, успешным ли было её выполнение;

Таким образом, в рамках данного исследования была определена оптимальная архитектура ПО для расширения полномочий веб-страниц: ПО должно быть приложением для операционной системы устройства пользователя, а веб-страница должна выполняться в обычном браузере, через который она и будет осуществлять взаимодействие с ПО для расширения полномочий. Для реализации ПО с данной архитектурой необходима разработка ПО ОС для расширения полномочий веб-страниц, а также компонента веб-страницы, отвечающего за взаимодействие с данным ПО, что позволит расширить полномочия веб-страницы.

## 2.2. Исследование возможности и способов определения уровня доверия к содержимому веб-страницы

### 2.2.1. Исследование возможности использования разработанного алгоритма для определения уровня доверия к содержимому веб-страницы

Вышеописанная архитектура, состоящая из компонента веб-страницы, взаимодействующего с сетью, а также из локального “серверного” приложения, отвечающего за работу с файлами, позволяет полностью выполнить задачу по расширению полномочий веб-страницы, но, при этом, не позволяет убедиться в том, что веб-приложение, которое получает столь значительные полномочия, является доверенным.

Если исходить из того, что приложение будет самостоятельно определять, является ли оно доверенным, это поставит под угрозу безопасность пользователя, так как вредоносное приложение заявит о том, что оно является доверенным. Это обстоятельство не позволяет доверять процесс определения того факта, является ли приложение доверенным, самому веб-приложению.

Соответственно, так как необходимо точно определять, является ли приложение доверенным, в разработанную архитектуру необходимо интегрировать список доверенных приложений.

Несмотря на то, что в данную архитектуру можно интегрировать данный список и предусмотреть процедуру проверки наличия приложения в данном списке. Даже при наличии аутентификации приложения какой-либо информации, которой располагают только разработчик конкретного приложения и администратор ПО для расширения полномочий веб-страниц, интеграция такого списка в ПО для расширения полномочий приведёт к следующим недостаткам системы:

- ПО для расширения полномочий является приложением, которое устанавливается на устройство пользователя. В случае, если в данном приложении будет содержаться список всех доверенных приложений, то его изменение потребует замены приложения у всех пользователей. В случае большого количества пользователей, сделать это будет крайне проблематично. При этом, при большом количестве приложений в данном списке, потребуется обновлять его у всех пользователей весьма часто, что также будет затруднительно, хоть и, в теории, возможно, но не позволит оперативно изменять список доверенных приложений;
- Из предыдущего недостатка также следует тот факт, что в распоряжении пользователя, хоть и не в открытом виде, будет находиться список доверенных приложений, что также является недостатком архитектуры, так как исключает возможность хранения в данном списке сведений, необходимых для аутентификации приложений, так как они будут доступны сразу всем разработчиком, а не определённым;
- Наиболее существенным недостатком, который следует из предыдущего пункта, является аутентификация приложения, так как, помимо того, что все сведения о приложениях будут доступны пользователям, все данные клиентской стороны веб-приложения также доступны пользователю

браузера. Из этого следует то, что веб-страница может быть полностью скопирована, и на основе безопасного и доверенного приложения может быть, с помощью копирования его исходного кода, создано вредоносное приложение, которое будет определяться ПО для расширения полномочий доверенных веб-страниц как доверенное;

На основании вышеописанных недостатков, <sup>2</sup> можно сделать вывод, что данная архитектура не является эффективной, и необходимо её изменить для исправления данных недостатков, с сохранением функционала, связанного с расширением полномочий веб-страниц.

#### 2.2.2. Определение перечня дополнительных компонентов ПО для определения уровня доверия к содержимому веб-страницы

Вышеописанные обстоятельства создают необходимость в интеграции в разработанную архитектуру дополнительных компонентов и приложений.

Прежде всего, для определения факта принадлежности приложения к реестру доверенных приложений, необходимо интегрировать в архитектуру базу данных (далее – БД), в которой будет храниться реестр приложений и выданных разрешений.

Для взаимодействия с БД с обеспечением безопасности и целостности данных, необходима также интеграция системы управления БД (далее – СУБД), которая сможет получать от единого серверного приложения, к которому будут обращаться приложения с устройств пользователей, команды для работы с БД, а также сообщать ему результат их выполнения.

Соответственно, с СУБД будет взаимодействовать единое серверное приложение, с которым будут взаимодействовать приложения, запущенные на устройствах пользователей. В рамках единого серверного приложения будет приниматься решение о том, имеет ли право конкретное веб-приложение на выполнение конкретной команды, то есть, будут происходить процессы аутентификации и авторизации команды. В рамках процесса аутентификации,

сервер принимает решение, была ли данная команда сформирована тем приложением, которое указано в передаваемых сведениях о команде. В рамках процесса авторизации, сервер принимает решение, разрешено ли данному приложению выполнять команды такого типа.

Так как необходимо вносить изменения в информацию, которая содержится в БД, то есть в список доверенных приложений, причём, как администрации системы в целях управления данным списком, так и сторонним разработчикам с целью добавления своих приложений в список, необходимо также приложение, позволяющее разработчикам и администрации в рамках своих полномочий работать с БД. Для максимальной простоты взаимодействия, данное приложение может быть разработано как веб-приложение, то есть иметь интерфейсы, как для администрации, так и для пользователей – разработчиков – размещённые на веб-страницах, и серверную часть, которая, являясь веб-сервером, будет получать запросы от веб-страниц с интерфейсами, и, после успешного осуществления аутентификации и авторизации, передавать команды пользователей СУБД для работы с БД.

Для того, чтобы приложение могло подтвердить, что конкретная команда была инициирована в рамках работы приложения на устройстве, сведения о команде и данных должно быть подписаны серверной компонентой веб-приложения, имеющего расширенные полномочия. Это позволит не включать в веб-приложение, запускаемое в браузере, секретный ключ приложения, так как исходный код веб-приложения может быть просмотрен в браузере пользователем, так как функция отображения исходного кода веб-страницы присутствует во всех популярных браузерах.

### 2.2.3. Составление алгоритма работы ПО для расширения полномочий доверенных веб-страниц

Таким образом, в рамках новой архитектуры, созданной с учётом исправления недостатков предыдущей, порядок действий для выполнения

команды с учётом проверки того, что команда выполняется доверенным приложением, у которого есть разрешение на выполнение конкретной команды, следующий (рисунок 7).

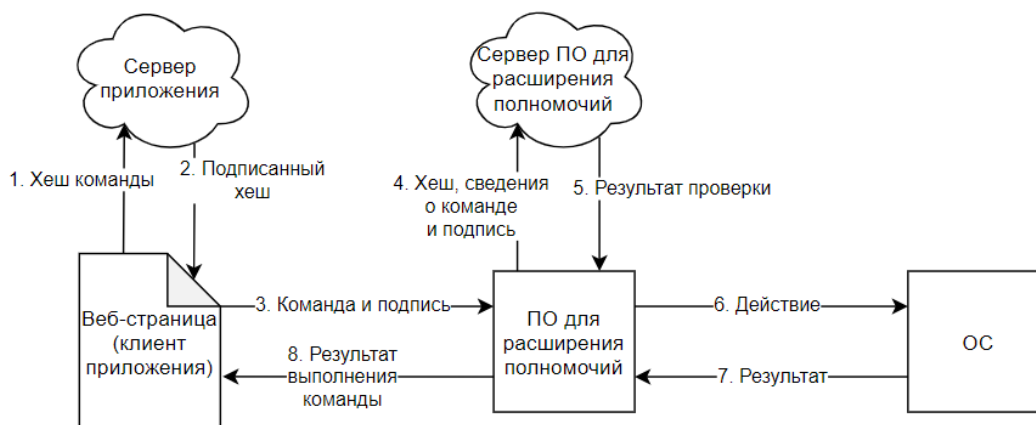


Рисунок 7. – Взаимодействие компонентов улучшенной архитектуры при выполнении команды

Как представлено на схеме на рисунке 7, в рамках данной архитектуры выполнение команды, инициируемой веб-страницей, происходит в следующие 8 этапов:

1. При наступлении некоторого события, определённого разработчиком, в веб-приложении, вызывается функция инициирования команды в библиотеке веб-приложения, которая отвечает за расширение его полномочий. В рамках данной библиотеки производится формирование объекта команды и его последующее хеширование. Хеш команды отправляется по сети серверу веб-приложения;
2. На сервере веб-приложения производится подпись хеша команды, полученного от клиента, с использованием секретного ключа приложения, который имеется в распоряжении у разработчика приложения. Результат подписи возвращается клиенту веб-приложения в качестве ответ на сетевой запрос, сделанный в пункте №1;
3. После получения ответа на сетевой запрос от своего сервера, веб-приложение отправляет запрос на выполнение команды, передавая приложению, расширяющему его полномочия все необходимые данные, а



- именно: объект команды, который участвовал в хешировании для сервера веб-приложения, а также подпись, которая была получена от сервера;
4. Приложение, отвечающее за расширение полномочий, передаёт хеш команды, её тип, подпись, полученную от веб-приложения и идентификатор приложения на свой сервер;
  5. На данном сервере происходит аналогичная подпись хеша команды с использованием секретного ключа приложения, который хранится в БД, с целью проверки секретного ключа, имеющегося в распоряжении у сервера веб-приложения, на равенство ключу, который хранится в БД сервера ПО расширения полномочий веб-страницы. Таким образом происходит процесс аутентификации, за которым следует процесс авторизации команды. Его результат, то есть, положительное или отрицательное решение о том, следует ли выполнять данную команду ПО на устройстве пользователя, возвращается ПО расширения полномочий в качестве ответа на сетевой запрос, отправленный в пункте №4;
  6. ПО для расширения полномочий, после получения положительного ответа от сервера, означающего, что выполнение команды разрешено, инициирует выполнение запрашиваемую команду на устройстве пользователя, отправляя соответствующий запрос на выполнение запрашиваемого действия ОС устройства пользователя;
  7. ОС возвращает ПО для расширения полномочий результат выполнения действия, инициированного в пункте №6;
  8. ПО, расширяющее полномочия веб-страницы, после получения результатов выполнения действия от ОС, формирует из них ответ для веб-страницы и отправляет его в качестве ответа на сетевой запрос, отправленный веб-страницей в пункте №3. Веб-приложение обрабатывает результат выполнения команды, что и позволяет пользователю взаимодействовать с веб-приложением с расширенными полномочиями;

Описанная выше архитектура позволяет обеспечить<sup>2</sup> расширение полномочий доверенных веб-страниц, в том числе, с учётом обеспечения проверки приложения на предмет доверия.

## 2.3. Проектирование программного обеспечения

2.3.1. Проектирование перечня компонентов системы расширения полномочий доверенных веб-страниц и технологий их реализации

В соответствии с поставленной задачей<sup>2</sup> и результатами проведённого исследования, была поставлена задача спроектировать систему расширения полномочий доверенных веб-страниц (далее – СРП), выбрать технологии реализации, а также разработать требования, предъявляемые с данной системе.

Для разработки СРП и тестирования работы данного ПО были спроектированы следующие компоненты:

- СРП, состоящая из четырёх компонентов:
  - Сервер СРП, отвечающий за аутентификацию и авторизацию иницилируемых команд. Сервер имеет известный Клиенту сетевой адрес, вследствие чего в рамках одной сети должен быть запущен строго один Сервер СРП, и данная часть СРП должна быть запущена на серверном сетевом устройстве всё необходимое для работы приложений, использующих СРП, время. В качестве технологий реализации выбраны язык программирования Python<sup>2</sup>, так как этот высокоуровневый язык[53, с.13] позволяет производить разработку с наименьшими трудозатратами, сосредотачиваясь на результате работы программы [54, с.1], Python-фреймворк Flask<sup>3</sup> для

---

<sup>2</sup> Документация Python [Интернет-ресурс] URL: <https://www.python.org/> (дата обращения: 06.05.2021)

<sup>3</sup> Документация Flask [Интернет-ресурс] URL: <https://flask.palletsprojects.com/en/1.1.x/> (дата обращения 06.05.2021)

работы с сетью по протоколу HTTP/1.1<sup>4</sup>, а также алгоритм HMAC, применяющийся [55, с.53], в том числе, для подписи SHA256<sup>5</sup> хеша команд в виде объектов, сериализованных в строки с использованием JSON<sup>6</sup>;

- Клиент СРП, то есть приложение, которое запущено на устройстве пользователя. Клиент должен быть запущен на устройстве пользователя всё то время, в которое требуется работа приложений, использующих СРП, для обеспечения выполнения инициируемых ими команд. В качестве технологий, также как и у Сервера СРП были выбраны язык Python, Python-фреймворк Flask и SHA256 для хеширования строковых представлений JSON-объектов. Для обмена данными с приложениями также используется протокол HTTP/1.1
- БД СРП, отвечающая за хранение сведений о приложениях. В качестве технологии реализации была выбрана MongoDB<sup>7</sup>, так как данная СУБД позволяет осуществлять разработку с максимальной скоростью [56, с. 961];
- Портал СРП, отвечающий за управление списком доверенных приложений и позволяющий разработчикам веб-приложений и администрации СРП иметь доступ, в рамках своих полномочий, к данному списку, хранящемуся в БД. В качестве технологий реализации были выбраны язык программирования Python и Python-фреймворк Flask, которые позволяют серверу данного веб-приложения взаимодействовать с клиентом по протоколу HTTP/1.1,

---

<sup>4</sup> Документация HTTP/1.1[Интернет-ресурс] URL: <https://tools.ietf.org/html/rfc2616> (дата обращения 06.05.2021)

<sup>5</sup> Алгоритм SHA256 [Интернет-ресурс] URL: <https://tools.ietf.org/html/rfc4634> (дата обращения 06.05.2021)

<sup>6</sup> Документация JSON[Интернет-ресурс] URL: <https://www.json.org/json-en.html> (дата обращения 06.05.2021)

<sup>7</sup> Документация MongoDB[Интернет-ресурс] URL: <https://www.mongodb.com/> (дата обращения 06.05.2021)

а также язык разметки HTML<sup>8</sup> в сочетании с языком программирования JavaScript<sup>9</sup>, языком разметки стилей CSS<sup>10</sup> и фреймворком Bootstrap<sup>11</sup>;

- Библиотека для использования в веб-приложении, формирующая запросы к Клиенту СРП. Данный компонент необходим по той причине, что его разработка позволит разработчикам веб-приложений иметь наиболее простой способ интеграции в СРП благодаря предоставляемому библиотекой функционалу. Это обеспечит возможность разработчику создавать только ту часть приложения, которая будет непосредственно выполнять прикладную задачу и взаимодействовать с пользователем. Такая часть приложения, соответственно, может быть представлена графическим интерфейсом с какой-либо автоматизацией, иницилирующим команды для СРП. В качестве технологии для реализации был выбран язык программирования JavaScript, в рамках которого применяются сетевые запросы по протоколу HTTP/1.1 и хеширование JSON-объектов с использованием технологии SHA256;
- Библиотека для использования в серверной части веб-приложений, использующих СРП. Данный компонент, также как и библиотека для использования в браузерной, то есть, клиентской части веб-приложения, позволит разработчику веб-приложения подключить библиотеку и сообщить ей необходимые сведения при вызове функции подписи, вместо того, чтобы разрабатывать приложение для подписи, соответствующее предъявляемым стандартам. В качестве технологий реализации был

---

<sup>8</sup> Документация HTML [Интернет-ресурс] URL: <https://www.w3.org/html/> (дата обращения 06.05.2021)

<sup>9</sup> Документация JavaScript [Интернет-ресурс] URL: <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/> (дата обращения 06.05.2021)

<sup>10</sup> Документация CSS [Интернет-ресурс] URL: <https://www.w3.org/Style/CSS/> (дата обращения 06.05.2021)

<sup>11</sup> Документация Bootstrap [Интернет-ресурс] URL: <https://getbootstrap.com/> (дата обращения 06.05.2021)

выбран язык Python и алгоритмы HMAC и SHA256 для подписи и хеширования JSON-объектов соответственно.

- Веб-приложение “Блокнот” – (далее – Блокнот), использующее СРП как независимое веб-приложение, зарегистрированное на Портале СРП. Данное веб-приложение необходимо для демонстрации и тестирования возможностей СРП в области взаимодействия с независимыми веб-приложениями, а также в области осуществления операций по чтению и записи текстовых файлов. В качестве технологий реализации были выбраны следующие: язык программирования Python, Python-фреймворк Flask для обмена с клиентом веб-приложения данными по протоколу HTTP/1.1, для реализации которого используются JavaScript, HTML, CSS и Bootstrap. Как клиент, и сервер данного приложения используют соответствующие библиотеки СРП.
- Веб-приложение “Проводник” (далее – Проводник), также использующее СРП как независимое веб-приложение, зарегистрированное на Портале СРП. В качестве технологий реализации были выбраны следующие: язык программирования Python, Python-фреймворк Flask для обмена с клиентом веб-приложения данными по протоколу HTTP/1.1, для реализации которого используются JavaScript, HTML, CSS и Bootstrap. Как клиент, и сервер данного приложения используют соответствующие библиотеки СРП. Данное приложение также позволяет протестировать функционирование СРП и продемонстрировать его возможности в области интеграции веб-приложений, а также в области осуществления операций, связанных с файловой системой, а именно:
  - Получение списка файлов в папке
  - Создание папок
  - Удаление файлов
  - Переименование файлов

Для разработки на языке Python было выбрано приложение Jupyter Notebook<sup>12</sup> – интерактивная оболочка, входящая в дистрибутив языка Anaconda<sup>13</sup>.

Соответственно, все компоненты, которые участвуют в выполнении каждой команды, которая выполняется СРП, имеют своё назначение, разработчика и пользователя.

### 2.3.2. Определение полномочий и задач пользователей системы

В качестве пользователей и разработчиков отдельных, и, в том числе, независимо разрабатываемых компонентов СРП имеются в виду следующие участники:

- Пользователь веб-приложения, использующий веб-приложение для решения своих прикладных задач. К таким задачам могут относиться, например, редактирование файлов или просмотр содержимого папок, далее – Пользователь. Участвия в разработке веб-приложений Пользователь не принимает, а лишь пользуется разработанными приложениями. Количество Пользователей может быть любым, так как любой человек может скачать Клиент СРП к себе на компьютер, и, таким образом, стать Пользователем. На устройстве Пользователя должны быть запущены браузер и Клиент СРП;
- Разработчик веб-приложения, которое выполняется в браузере на устройстве Пользователя и использует СРП для работы с устройством Пользователя – Разработчик. Для того, чтобы приложение было добавлено в список доверенных приложений, Разработчик производит необходимые для этого действия, связываясь с администрацией СРП

---

<sup>12</sup> Документация Jupyter [Интернет-ресурс] URL: <https://jupyter.org/> (дата обращения 06.05.2021)

<sup>13</sup> Описание Anaconda [Интернет-ресурс] URL: <https://www.anaconda.com/> (дата обращения 06.05.2021)

через Портал СРП. Также, Разработчик обеспечивает возможность работы веб-приложения в браузере на устройстве Пользователя и работу сервера веб-приложения. В рамках Портала СРП, Разработчик является пользователем Портала СРП с ограниченными полномочиями. Соответственно, количество разработчиков может быть любым, так как любой человек может стать разработчиком и подать заявку на регистрацию приложения через Портал СРП.

- Администратор СРП – разработчик всех компонентов СРП и пользователь Портала СРП с полными полномочиями в управлении списком доверенных веб-приложений. Список Администраторов СРП определяется непосредственно пользователем административных компонентов СРП, то есть Сервера СРП и Портала СРП.

Соответственно, у каждого из компонентов СРП имеются свои назначение, пользователь и разработчик. Это крайне важно рассмотреть как проектирования, так и для дальнейшей эксплуатации СРП. Данная информация отражена в следующей таблице (таблица 11).

Таблица 11. Компоненты СРП

Компонент	Назначение	Пользователь	Разработчик
Портал СРП	Управление списком доверенных приложений Администраторами СРП и Разработчиками в рамках полномочий	Разработчик и Администратор СРП	Администратор СРП
Клиент СРП	Взаимодействие веб-приложений, использующих СРП с устройством пользователя	Пользователь	Администратор СРП

Сервер СРП	Работа со списком доверенных приложений	Администратор СРП	Администратор СРП
Веб-приложение (в том числе, Блокнот и Проводник)	Выполнение прикладных задач	Пользователь	Разработчик
Сервер веб-приложения	Подтверждение безопасности команд, инициируемых веб-приложением	Разработчик	Разработчик
Клиентская библиотека СРП	Взаимодействие веб-приложения с Клиентом СРП	Разработчик (при разработке веб-приложения)	Администратор СРП
Серверная библиотека СРП	Взаимодействие сервера веб-приложения с Клиентской библиотекой СРП	Разработчик	Администратор СРП

Как представлено в таблице 11, выполнение каждой команды возможно только при участии всех трёх участников взаимодействия: Пользователь должен получить от Администратора Клиент СРП, после чего получить от Разработчика веб-приложение. Разработчик должен получить клиентскую и серверную библиотеки от Администратора СРП, и предоставить Пользователю веб-приложение, поддерживать работу сервера веб-приложения и взаимодействовать через Портал СРП. Администратор СРП должен поддерживать работу Сервера СРП и Портала СРП, предоставлять Пользователю Клиент СРП, и взаимодействовать с Разработчиком через Портал СРП.

### 2.3.3. Формирование требований к системе

После того, как, в соответствии с результатом исследования, был определён функционал ПО для расширения полномочий доверенных веб-



страниц, необходимо было также определить следующие требования, которые предъявлялись к СРП:

- Все компоненты СРП не должны при использовании аварийно завершаться;
- Все компоненты СРП должны функционировать таким образом, чтобы выполнялся функционал их назначения, описываемый в Таблице 11;
- Клиент СРП должен обеспечивать расширение полномочий только доверенных веб-страниц;
- Сервер СРП должен осуществлять аутентификацию и авторизацию всех команд;
- Портал СРП должен быть доступен как Пользователям и Разработчикам, так и Администраторам;
- Только Администраторы могут вносить изменения в БД СРП через Портал, работающий в режиме полномочий администратора;
- Только Разработчик конкретного приложения может осуществлять управление им на Портале СРП в рамках своих полномочий.

Соответственно, СРП была разработана в соответствии с разработанной архитектурой, выбранным технологическим стеком и предъявленными требованиями.

## ГЛАВА 3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

### 3.1. Разработка программного обеспечения для определения уровня доверия к содержимому веб-страницы

#### 3.1.1. Проектирование базы данных СРП

В соответствии с тем, что в качестве технологии реализации БД была выбрана СУБД MongoDB, поддерживающая, в том числе, хранение строковых данных, а также данных о дате и времени, были спроектированы поля двух объектов:

- Объект приложения, предназначенный для хранения сведений о приложении, хранящийся в коллекции apps (таблица 12).

Таблица 12. Поля объекта приложения

Имя поля	Назначение	Тип
_id	Идентификатор объекта	Идентификатор
name	Наименование приложения	Строка
desc	Описание приложения	Строка
email	Адрес электронной почты разработчика	Строка
devName	Наименование разработчика	Строка
list	Разрешение на выполнение команды для получения списка файлов и папок	Строка
read	Разрешение на выполнение команды чтения	Строка
delete	Разрешение на выполнение команды удаления	Строка
mkdir	Разрешение на выполнение команды создания папки	Строка
rename	Разрешение на выполнение команды переименования	Строка

salt	“Соль” – случайные данные для хеширования пароля	Строка
passwordHash	Хеш пароля и “соли”	Строка
secretKey	Секретный ключ приложения	Строка
date	Дата регистрации приложения	Дата и время
appSite	Адрес веб-приложения в интернете	Строка

Поля объекта приложения соответствуют перечисленным полям в таблице 12.

- Объект временной ссылки для выдачи секретного ключа, предназначенный для хранения в коллекции links (таблица 13).

*Таблица 13 – поля объекта временной ссылки*

Имя поля	Назначение	Тип
_id	Идентификатор объекта	Идентификатор
appID	Идентификатор приложения	Идентификатор

Поля объекта временной ссылки соответствуют перечисленным полям в таблице 13.

Для имени спроектированной базы данных было выбрано значение “srp”.

### 3.1.2. Разработка Сервера СРП

Сервер СРП – приложение, взаимодействующее с Клиентом СРП и БД СРП для осуществления проверки команд, которые инициируют приложения.

Так как данный сервер реализуется на языке программирования Python с использованием фреймворка Flask, в начале его работы инициализируется объект сервера Flask:

```
app = Flask(__name__)
```

После того, как Сервер СРП получает от Клиента СРП команду для проверки, он производит проверку следующим образом:

Под фактом проверки команды было выбрано считать факт, что команда была инициирована в результате выполнения следующих условий:

- Веб-приложение имеет действующее разрешение на запрашиваемое в команде действие
- Веб-приложение было загружено с сайта его разработчика

Определение факта выполнения данных условий позволяет убедиться в том, что команда была не была инициирована в рамках атаки, проводимой с целью несанкционированного использования Клиента СРП, запущенного на компьютере пользователя.

Для того, чтобы убедиться в том, что веб-приложение было загружено с сайта его разработчика, необходима проверка адреса веб-страницы: данная проверка должна происходить на Сервере СРП, а не в Клиенте СРП, с целью обеспечения актуальности этих данных, а также с целью обеспечения наибольшей степени доверия к данному списку и к результатам проверки. Данная проверка производится следующим образом:

```
if request.json["appURL"]!=appObject["appSite"]:
    return "False"
```

Соответственно, если адрес не совпадает, команда окончательно считается не прошедшей проверку, из-за чего отрицательный результат проверки незамедлительно сообщается Клиенту СРП.

Проверка команды производится в два этапа, выполняющихся в последовательном порядке:

1. Аутентификация команды
2. Авторизация команды

Для выполнения процесса аутентификации, веб-приложение производит хеширование JSON-объекта, имеющего следующие поля:

- type – строка, содержащая тип инициируемой команды;
- args – объект, содержащий в себе все аргументы, необходимые команде для выполнения;

Перед осуществлением хеширования, веб-приложение выполняет сортировку полей объекта в алфавитном порядке, а также такую же сортировку

в рамках вложенного объекта `args`. Данная сортировка необходима, чтобы результат хеширования совпадал при выполнении хеширования на нескольких устройствах. <sup>2</sup> Это связано с тем, что по умолчанию объект, как тип данных, не содержит определённого порядка следования полей, вследствие чего может произойти ситуация, в которой при фактически одинаковых объектах будет наблюдаться различный результат хеширования в различных приложениях из-за несовпадающего порядка полей объекта.

Результат данного хеширования отправляется на сервер веб-приложения, где выполняется следующий шаг аутентификации. Он заключается в использовании секретного ключа приложения для подписи хеша. Использование типа команды и аргументов для её выполнения вместо хеша, безусловно, обеспечивало бы ещё более высокий уровень безопасности, однако применение хеширования, в данном случае, обусловлено следующими обстоятельствами:

- Аргументы команды могут иметь весьма большой объём: в случае, например, записи файла большого объёма, необходимо включить в аргументы команды все данные, которые необходимо записать в память, из-за чего размер вышеописанного объекта не может быть меньше, чем размер файла, который будет создан в результате выполнения команды. Данное обстоятельство не оказывает негативного влияния на эффективность работы системы в части взаимодействия веб-приложения и Клиента СРП, но отправка больших объёмов информации на сервер веб-приложения с большой периодичностью может создать чрезмерную нагрузку на интернет-соединения как устройства пользователя, так и самого сервера веб-приложения, что может значительно снизить эффективность СРП, а также ухудшит эффективность многих приложений на пользовательском устройстве;
- Содержимое записываемых файлов (а также, например, названия удаляемых файлов) будет передано на сервер веб-приложения, из-за чего

данный сервер будет иметь доступ ко всей информации, которую пользователь будет сохранять на своём устройстве. В то время как при передаче команды в Клиент СРП хранения, анализа и распространения данной информации не будет производиться, что явно можно проконтролировать при разработке СРП, у сервера веб-приложения появится возможность производить такие действия. Данное отсутствие приватности, очевидно, недопустимо при функционировании системы; Осуществление данной операции на стороне сервера обусловлено следующими обстоятельствами:

- Секретный ключ не встраивается в исходный код клиентской части веб-приложения. Это не позволяет использовать даже полную копию исходного кода веб-приложения, которая доступна любому пользователю в браузере для выполнения команд от имени зарегистрированного доверенного приложения;
- При осуществлении данного запроса на сервер веб-приложения, сервер может убедиться в правильности используемого приложением хоста. Это также не позволит разместить копию зарегистрированного приложения по какому-либо другому адресу в интернете, так как сервер осуществляет проверку хоста приложения. Если не выполнять данную проверку, это создаст риск атаки на компьютер пользователя с использованием СРП: так как клиентский код веб-приложения, передаваемый браузеру, доступен пользователям, то полностью скопировать клиентскую часть веб-приложения является возможным, также как и дальнейшее размещение точно такой же клиентской части на другом сайте. При размещении копии может быть изменён функционал без изменения внешнего вида веб-приложения, что может быть выгодно разработчику “копии” веб-приложения и невыгодно пользователю. Такое приложение может быть создано как вредоносная программа, анализирующая или удаляющая файлы с устройства пользователя вне зависимости от желания

пользователя, но с использованием сервера “настоящего” веб-приложения;

Результат подписи отправляется веб-приложению в качестве ответа на HTTP-запрос. Таким образом, после успешного получения ответа на данный запрос от сервера, у веб-приложения имеются следующие данные, относящиеся к инициированию конкретной команды:

- Идентификатор приложения;
- Тип команды;
- Аргументы команды;
- Подпись хеша команды серверной стороной приложения;

Так этих данных достаточно как для выполнения иницируемой команды, так и для подтверждения того факта, что программа была иницирована в рамках доверенного приложения, веб-приложение отправляет Клиенту СРП сетевой запрос по сети внутри устройства пользователя, в котором сообщает всю необходимую информацию для безопасного инициирования команды в виде объекта:

- `appID` – строка, содержащая в себе идентификатор веб-приложения;
- `serverSign` – строка, содержащая в себе подпись хеша команды на сервере веб-приложения;
- `type` – строка, содержащая в себе тип иницируемой команды;
- `args` – объект, содержащий в себе все аргументы, которые необходимы команде данного типа для выполнения;

После того, как Клиент СРП получает вышеприведённые данные, которых достаточно для выполнения команды, для продолжения процесса аутентификации команды, Клиент СРП отправляет сетевой запрос на Сервер СРП, в котором передаёт Серверу СРП следующий объект:

- `appID` – строка, содержащая в себе идентификатор веб-приложения;
- `serverSign` – строка, содержащая в себе подпись команды, полученную от сервера веб-приложения;

- `type` – строка, содержащая в себе тип иницилируемой команды;
- `commandHash` — строка, содержащая результат хеширования команды, то есть следующего объекта:
  - `type` – строка, содержащая тип иницилируемой команды;
  - `args` – объект, содержащий в себе все аргументы, необходимые команде для выполнения;

Перед осуществлением хеширования для получения значения переменной `commandHash`, Клиент СРП осуществляет сортировку полей вышеописанного объекта по той же причине, по которой осуществлялась данная сортировка на стороне веб-приложения.

Осуществление данного хеширования позволяет избежать многих недостатков при работе системы, по следующим причинам:

- Сервер СРП не получает данные, с которыми работает Пользователь. Этот подход способствует максимальной безопасности данных Пользователя, так как они не выходят за пределы его устройства;
- Количество данных, отправляемых по сети, значительно сокращается, что способствует снижению нагрузки на сетевое подключение Пользователя, на сетевое подключение Сервера СРП, а также на сам Сервер СРП;

Когда Сервер СРП получает запрос от Клиента СРП, он производит следующие операции:

1. Извлечение из БД СРП секретного ключа приложения
 

```
appObject=db.apps.find_one({"_id":ObjectId(request.json["appID"])})
appObject["secretKey"]
```
2. Подпись хеша команды, полученного от Клиента СРП с использованием секретного ключа приложения из БД СРП
 

```
b64encode(HMAC(appObject["secretKey"].encode(),
request.json["commandHash"].encode(), sha256).digest())
```
3. Сравнение подписи из пункта №2 со значением переменной `serverSign`, которая была получена от Клиента СРП:



```
tSign.decode("utf-8")==request.json["serverSign"]
```

В случае, если в пункте №3 значения переменных совпадают, можно считать, что аутентификация команды выполнена успешно.

В случае, если данное сравнение показывает, что результаты хеширований не совпадают, из этого следует, что подпись на стороне сервера веб-приложения производилась с использованием другого секретного ключа или хеша другой команды. Так как можно считать достоверными сведения из БД СРП, это означает, что на сервере веб-приложения находится недействительный секретный ключ приложения, из-за чего невозможно считать команду инициированной доверенным приложением. Данная ситуация может произойти при попытке выдать приложение, не включённое в список доверенных приложений, за доверенное. В этой ситуации процесс аутентификации команды следует считать завершённым неуспешно.

Результат процесса аутентификации, в том числе, возвращается Клиенту СРП в качестве ответа на сетевой запрос. В том случае, если Клиент СРП получает от Сервера СРП ответ о неуспешном завершении процесса аутентификации, Клиент СРП не осуществляет выполнение инициированной веб-приложением команды.

Помимо процесса аутентификации команды, крайне важно осуществлять на Сервере СРП процесс авторизации команды. То есть, определять, разрешено ли веб-приложению использовать команду инициированного типа. Например, приложение для просмотра файлов может не иметь полномочий на удаление файлов с устройства пользователя. Помимо этого, необходимо убедиться в том, что деятельность приложения на текущий момент одобрена администрацией СРП.

Процесс авторизации команды также, как и процесс аутентификации, должен проходить на стороне Сервера СРП, так как единственные достоверные сведения о том, команды какого типа разрешены соответствующему приложению, хранятся в БД СРП.

Для осуществления процесса авторизации, Сервер СРП выполняет следующие действия:

1. Извлечение из БД списка типов команд, которые приложение имеет право использовать на текущий момент по решению администрации СРП:

```
allowedCommands=list()
for i in ["list","read","write","delete","mkdir","rename"]:
    if appObject[i]=="on":
        allowedCommands.append(i)
```

2. Поиск в списке из пункта №1 типа команды, который запрашивается в иницилируемой команде

```
request.json["type"] in allowedCommands
```

В том случае, если результат поиска свидетельствует о наличии в списке разрешённых типов команд типа иницилируемой веб-приложением команды, можно считать процесс авторизации завершённым успешно.

В случае, если найти в списке разрешённых для выполнения типов команд, найти тип иницилируемой команды невозможно, следует считать процесс авторизации команды завершённым неуспешно.

Только в том случае, если и процесс аутентификации, и процесс авторизации были завершены Сервером СРП успешно, Клиенту СРП необходимо выполнить команду, которая была инициирована веб-приложением.

Клиенту, соответственно, возвращается общий результат аутентификации, который может принимать два строковых значения:

- True – аутентификация и авторизация пройдены успешно;
- False – аутентификация или авторизация не были завершены успешно;

После того, как объекту сервера сообщены все функции для взаимодействия, в нём вызывается метод запуска:

```
app.run(host=SERVER_HOST, port=SERVER_PORT, threaded=True)
```

Файл исходного кода Сервера СРП “Server SRP.ipynb” находится в репозитории в папке Server SRP.

### 3.1.3. Разработка Портала СРП

Для того, чтобы Администраторы и Разработчики, а также Пользователи могли иметь доступ к списку приложений, хранящемуся в БД СРП, был разработан Портал СРП, представляющий из себя веб-приложение.

Также как и Сервер СРП, данное приложение было разработано с использованием фреймворка Flask. У объект сервера, соответственно, вызывается метод для запуска веб-сервера:

```
app.run(host=PORTAL_HOST, port=PORTAL_PORT, threaded=True)
```

Для описания работы портала, необходимо рассмотреть работу с ним Пользователя, Разработчика и Администратора.

Пользователю, для доступа к Порталу СРП, достаточно открыть сайт без дополнительного адреса страницы, и ему откроется следующая страница (рисунок8).

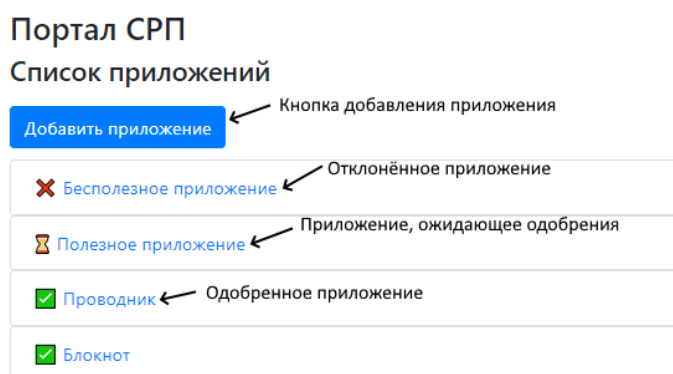


Рисунок 8. – Пример начальной страницы для пользователя

Как продемонстрировано на рисунке 8, вместе с кнопкой для добавления приложения, на страницу выводится также список приложений, отсортированных по убыванию по дате регистрации. При этом, приложения в списке обозначены как:

- Отклонённые – в случае, если приложению не разрешено выполнение ни одного типа команды. Приложения переводятся в данную категорию из “ожидающих” или “одобренных” по решению Администратора;
- Ожидающие одобрения – в случае, если у команд, которые выбрал Разработчик в форме добавления приложения, в БД СРП установлено значение “запрос”. Все добавленные приложения автоматически становятся “ожидающими одобрения”;
- Одобренные – в случае, если приложению разрешено выполнение хотя бы одного типа команды. Приложения переводятся в данную категорию из “ожидающих” или “отклонённых” по решению Администратора;

Данный список формируется из объектов в коллекции apps в БД СРП.

Если Пользователь нажмёт на название приложения, то он перейдёт на страницу информации о приложении, на которой отображена следующая информация о приложении, полученная из объекта с соответствующим идентификатором (идентификатором приложения), хранящемуся в коллекции apps в БД СРП:

- Идентификатор приложения;
- Наименование приложения;
- Наименование разработчика;
- E-mail разработчика;
- Ссылка на приложение, перейдя по которой можно воспользоваться приложением;
- Описание приложения;
- Дата и время регистрации приложения;
- Статус разрешения на использование команд:
  - Получения списка файлов;
  - Чтения файла;
  - Записи файла;
  - Удаления файла;

- Создания папки;
- Переименования;

Соответственно, строковые поля информации о приложении представлены следующим образом (рисунок 9).

**Портал СРП**  
**Приложение**

[На главную](#) ← Возврат к списку приложений

Идентификатор приложения

60905eb3856012aac75110e2

Уникальный идентификатор приложения в системе

Наименование приложения

Полезное приложение

Наименование разработчика

Иванов И.И.

E-mail разработчика

mail@example.com

Ссылка на приложение

<https://example.com>

Описание приложения

Полезное приложение, выполняющее полезную функцию

Дата регистрации (UTC)

2021-05-03 20:36:03.757000

Рисунок 9. – Пример строковых полей информации о приложении

Как продемонстрировано на рисунке 9, строковые поля из вышеприведённого списка отображены в виде нередактируемых текстовых полей. Ссылка на приложение представлена в виде ссылки, чтобы пользователь Портала СРП мог по ней перейти к приложению с помощью нажатия. Также, на данной странице имеется кнопка возврата к списку приложений.

Статусы разрешения на использование приложением команд представлены на той же странице следующим образом (рисунок 10).

Список файлов Запрошено ▾

Чтение файла Недоступно ▾

Запись в файл Недоступно ▾

Удаление файла Запрошено ▾

Создание папки Недоступно ▾

Переименование Недоступно ▾

Для разработчика

[Управление приложением](#)

Рисунок 10. – Пример статусов разрешения на использование приложением команд

Как показано на рисунке 10, напротив каждого типа команды показан текущий статус, а ниже располагается кнопка “Управление приложением”, предназначенная для Разработчика приложения, позволяющая перейти на панель управления приложением. Для Пользователей, функционал Портала СРП ограничен данными страницами, так как им доступен только просмотр информации о приложения. При этом, информация, доступная для просмотра, не включает в себя информацию, связанную с аутентификацией приложения, то есть хеш пароля и секретный ключ.

Для Разработчиков, прежде всего, предназначена страница добавления приложения.

В её первой части, Разработчик заполняет форму следующим образом (рисунок 11).

# Портал СРП

## Добавление приложения

[На главную](#)

Наименование приложения

Полезное приложение

Поле не может быть пустым

Наименование разработчика

Иванов И.И.

Поле не может быть пустым

E-mail разработчика

mail@example.com

Поле не может быть пустым

Ссылка на приложение

https://example.com

Поле не может быть пустым

Описание приложения

Полезное приложение, выполняющее полезную функцию

Администрация принимает решения об одобрении приложения на основании его названия и описания

Рисунок 11. – Пример заполнения разработчиком формы добавления приложения

Как представлено на рисунке 11, для добавления приложения на Портал СРП, разработчику необходимо заполнить соответствующие поля, связанные с приложением. При этом, в данной форме используется валидация полей, из-за чего нельзя отправить на сервер Портала СРП форму, в которой не заполнено какое-либо поле.

Помимо данных полей, Разработчику также необходимо заполнить дополнительные поля, непосредственно связанные с работой приложения (рисунок 12).

Разрешения, необходимые приложению:

☒ Список файлов

Получение списка файлов и папок в папке по её абсолютному пути

☐ Чтение файла

Получение содержимого файла по его абсолютному пути

☐ Запись в файл

Запись содержимого в файл по его абсолютному пути

☒ Удаление файла

Удаление файла по его абсолютному пути

☐ Создание папки

Создание папки по её абсолютному пути

☐ Переименование

Переименование файла или папки по абсолютному пути и новому имени

Пароль разработчика приложения

.....

Минимум 10 символов

Добавить приложение

Рисунок 12. – Пример заполнения разработчиком второй части формы добавления приложения

На рисунке 12 продемонстрировано, как Разработчик, отмечая галочками, выбирает, какие типы команд будет использовать его приложение, а также вводит пароль, который впоследствии позволит Разработчику управлять приложением в рамках своих полномочий. Длина пароля также проходит валидацию в рамках данной веб-страницы. Помимо этого, все поля, которые должны быть заполнены, проходят дополнительную проверку на северной стороне портала, например:

```
if request.form.get('appSite', '')=='':  
    return redirect("/createError")
```

То есть, если при отправке формы, её данные не будут корректными, сервер не произведёт добавление приложения, а переадресует Разработчика на страницу ошибки добавления. При успешном добавлении, на сервере происходит формирование объекта приложения, включающее в себя



хеширование пароля и генерацию “соли”, и добавление его в коллекцию apps БД СРП, а также, в коллекцию links добавляется объект временной ссылки, а Разработчик переадресуется на страницу, на которой показан секретный ключ приложения (рисунок 13).

## Портал СРП

### Секретный ключ приложения

[На главную](#)

Идентификатор приложения

60906298856012aac75110eb

Уникальный идентификатор приложения в системе

Секретный ключ приложения

f85e172e54d99f87edc140810971f895e077315e7b80acfbfeb3e75c5cf38d8f

Сохраните его для работы с системой

[К приложению](#)

Рисунок 13. – Пример страницы отображения секретного ключа приложения

Страница, представленная на рисунке 13, показывается, когда с помощью адреса страницы, содержащего идентификатор временной ссылки, сервер находит в коллекции links данную ссылку, извлекает из коллекции apps соответствующее приложение, находя его по идентификатору, и, получив, таким образом, секретный ключ, удаляет объект открытой ссылки из коллекции links. Таким образом, секретный ключ отображается только один раз, повторно Разработчик может получить секретный ключ только при запросе нового секретного ключа на странице управления приложением при вводе пароля разработчика (рисунок 14).

## Портал СРП

### Управление приложением

[На главную](#)

Идентификатор приложения

60905eb3856012aac75110e2

Уникальный идентификатор приложения в системе

Наименование приложения

Полезное приложение

[К приложению](#)

#### Генерация нового секретного ключа

Пароль разработчика

Введите пароль

[Сгенерировать новый секретный ключ](#)

#### Изменение пароля разработчика

Пароль разработчика

Введите пароль

Новый пароль разработчика

Введите новый пароль

Минимум 10 символов

[Изменить пароль](#)

#### Удаление приложения

Пароль разработчика

Введите пароль

[Удалить приложение](#)

Рисунок 14. – пример страницы управления приложением

Как показано на рисунке 14, Разработчик, используя пароль разработчика конкретного приложения, может его изменить, сгенерировать новый секретный ключ, а также удалить приложение с Портала СРП. Соответственно, для каждой команды существует отдельная форма с полем для пароля и кнопкой действия. При инициировании данных действий, сервер

Портала СРП проверяет, что пароль разработчика правильный. Так как в БД СРП не хранятся пароли Разработчиков, сервер Портала СРП производит хеширование пароля вместе с его “солью”, которая хранится в объекте приложения в БД СРП, также как и правильный хеш пароля, с которым и сравнивается полученный сервером Портала СРП хеш:

```
sha256((salt+request.form.get('devPassword','')).encode('utf-8')).hexdigest()
```

Данные страницы доступны для использования Разработчикам, и позволяют разработчику каждого приложения управлять своим приложением, при условии, что разработчик знает правильный пароль от приложения.

Для того, чтобы войти в режим администрирования списка приложений, Администратор открывает страницу по адресу “/admin” на сайте Портала СРП, после чего ему отображается список приложений (рисунок 15).

## Портал СРП (для администраторов)

### Список приложений

Добавить приложение

✗	Бесполезное приложение
⌚	Полезное приложение
✓	Проводник
✓	Блокнот

Рисунок 15. – пример списка приложений в режиме администрирования

На рисунке 15 показан пример списка приложений в режиме администрирования. Маркировка приложений аналогична версии списка для обычных пользователей. При нажатии на название откроется страница приложения в режиме администрирования (рисунок 16).

## Портал СРП (для администраторов)

### Приложение

[На главную](#)

Идентификатор приложения

60905eb3856012aac75110e2

Уникальный идентификатор приложения в системе

Наименование приложения

Полезное приложение

Наименование разработчика

Иванов И.И.

E-mail разработчика

mail@example.com

Ссылка на приложение

https://example.com

Описание приложения

Полезное приложение, выполняющее полезную функцию

Дата регистрации (UTC)

2021-05-03 20:36:03.757000

Рисунок 16. – пример части страницы приложения в режиме администрирования

Как показано на рисунке 16, в части страницы редактирования приложения Администратором, посвящённой строковым значениям, есть редактируемые и нередактируемые текстовые поля, так как Администратор не может вносить изменения в идентификатор приложения и в дату его создания. На данной странице Администратор также может редактировать разрешения приложения на выполнение команд, пароль разработчика приложения,

генерировать новый секретный ключ, а также осуществить удаление приложения (рисунок 17).

Список файлов Запрошено ▾

Чтение файла Недоступно ▾

Запись в файл Недоступно ▾

Удаление файла Запрошено ▾

Создание папки Недоступно ▾

Переименование Недоступно ▾

Пароль администратора

**Сохранить изменения**

**Изменение пароля разработчика**

Пароль администратора

Новый пароль разработчика

Минимум 10 символов

**Изменить пароль**

**Удаление приложения**

Пароль администратора

**Удалить приложение**

Рисунок 17. – Пример формы управления приложением Администратором

Как продемонстрировано на рисунке 17, Администратор может внести изменения в объект приложения в списке, но только при вводе в соответствующую форму правильного пароля администратора. Пароль администратора не включён в исходный код сервера Портала СРП, а хранится в отдельном файле в папке сервера. В случае, если данного файла нет в данной папке при запуске сервера Портала СРП, он автоматически генерируется со стандартным значением пароля Администратора:

```
if os.path.isfile('.ADMIN_PASSWORD'):
    with open('.ADMIN_PASSWORD','r') as f:
        ADMIN_PASSWORD = f.read()
else:
    with open('.ADMIN_PASSWORD','w') as f:
```

```
ADMIN_PASSWORD = "ADMIN_PASSWORD"  
f.write(ADMIN_PASSWORD)
```

Только с помощью редактирования приложения Администратором, приложению может быть одобрено использование команд.

В Портале СРП используется функционал шаблонов Flask, позволяющий передавать HTML-страницам значения из кода программы, например, при просмотре информации о приложении, при формировании страницы используется переменная `appObject`, идентификатор из которой встраивается Flask в код страницы на серверной стороне Портала СРП:

```
value="{{ appObject._id }}"
```

Файлы исходного кода Портала СРП расположены в папке Portal SRP:

- Portal SRP.ipynb – файл исходного кода сервера Портала СРП;
- templates – папка с исходным кодом страниц портала:
  - adminApp.html – страница редактирования приложения;
  - appList.html – страница списка приложений;
  - createError.html – страница ошибки при добавлении приложения;
  - devApp.html – страница управления приложением Разработчиком;
  - formCreate.html – страница с формой добавления приложения;
  - getApp.html – страница просмотра информации о приложении;
  - saveError.html – страница ошибки при изменении приложения;
  - secretKey.html – страница отображения секретного ключа

### 3.1.4. Разработка Серверной библиотеки СРП

Серверная библиотека СРП предназначена для Разработчиков. Она позволяет серверу приложения только подключить её и вызвать для генерации НМАС-подписи хеша.

Библиотека представляет из себя одну функцию – `signSRP`, принимающую два аргумента:

- Секретный ключ приложения;
- Хеш команды, полученный от Клиентской библиотеки СРП;

Данная функция возвращает строку подписи для передачи её Клиентской библиотеке СРП.

Файл исходного кода Серверной библиотеки СРП “`appServerSPP.py`” находится в репозитории в папке `ServerLib SRP`.

## 3.2. Разработка программного обеспечения для расширения полномочий доверенных веб-страниц

### 3.2.1. Разработка Клиента СРП

Клиент СРП является компонентом СРП, который непосредственно отвечает за расширение полномочий веб-страниц, которые, благодаря его взаимодействию с Сервером СРП, можно считать доверенными. Клиент СРП должен быть запущен на компьютере Пользователя всё то время, что Пользователь собирается использовать приложения, использующие СРП. Клиент СРП, также, как и другие компоненты, написан на языке программирования Python и использует фреймворк Flask для взаимодействия с приложениями и Сервером СРП по протоколу HTTP/1.1.

Соответственно, сервер Клиента запускается после инициализации его функции:

```
app.run(host=CLIENT_HOST, port=CLIENT_PORT, threaded=True)
```

Функция, которая отвечает на запросы Клиентской библиотеки СРП и отправляет запросы Серверу СРП, работает следующим образом:

1. Получает команду;
2. Вычисляет SHA256-хеш команды:  
`sha256(commmandInJSON.encode("utf-8")).hexdigest()`
3. Отправляет сведения о команде на Сервер СРП:  
`requests.post(SERVER_URL, json = commandObject)`
4. В случае получения разрешения на выполнение команды от сервера, происходит её выполнение;
5. После выполнения команды, её результат или факт ошибки при выполнении возвращается Клиентской библиотеке СРП;

Выполнение команды происходит следующим образом:

- Чтение файла: попытка чтения файла и преобразование его содержимого в список байтов;
- Запись в файл: попытка записи списка байт в файл по указанному пути;
- Получение списка файлов: попытка получения сведений о всех файлах и папках, содержащихся в запрашиваемой папке. Пустой путь обрабатывается как получение списка дисков, подключенных к устройству пользователя. В случае, если вместо пути используется переменная окружения, она интерпретируется как соответствующий ей в ОС путь;
- Удаление: проверка типа объекта: если это файл, то происходит попытка удаления файла или папки;
- Создание папки: попытка создания папки по указанному пути;
- Переименование: попытка переименования файла или папки:  
`os.rename(path, newName);`

Для того, чтобы работать со всеми символами, которые могут быть в пути, например, с кириллическими символами, используется функции `quote` и `unquote` библиотеки `Urllib`, входящей в язык `Python`. Если при выполнении операции происходит ошибка, то есть, ОС не может выполнить команду успешно, это обрабатывается Клиентом СРП, и об этом сообщается



приложению. Если Сервер СРП не разрешает выполнение команды, Клиент СРП не выполняет её, и сообщает об этом приложению.

Файл исходного кода Клиента СРП “Client SRP.ipynb” находится в репозитории в папке Client SRP.

### 3.2.2. Разработка Клиентской библиотеки СРП

Клиентская библиотека, написанная на языке JavaScript, предназначена для взаимодействия веб-страницы с Клиентом СРП и с Серверной библиотекой СРП. Библиотека позволяет Разработчику подключить её к веб-странице для упрощения работы с СРП.

Для того, чтобы запустить команду, Разработчику необходимо запустить функцию `runSRPCommand`, которая принимает следующие аргументы:

- `appID` – идентификатор приложения;
- `type` – тип иницируемой команды;
- `args` – объект, содержащий в себе аргументы команды;
- `signURL` – адрес, по которому можно осуществить подпись хеша команды;
- `onWaiting` – функция, которая будет вызвана во время ожидания завершения команды, аргументы не передаются;
- `onDone` – функция, которая будет вызвана при успешном завершении команды, аргументы не передаются;
- `onError` – функция, которая будет вызвана при неуспешном завершении команды, аргументы не передаются;
- `onReturn` – функция, которая будет вызвана при успешном завершении, в качестве аргументов передаются объект результата выполнения команды и объект команды;

При вызове функции происходит вызов функции `onWaiting` без передачи аргументов. Затем, происходит замена всех обратных слешей на прямые для упрощения работы с файловой системой:

```
encodeURIComponent(args.path.replaceAll("\\", "/"))
```

После этого формируется объект команды. После сортировки объекта команды, вычисляется её хеш для подписи Серверной Библиотекой СРП. После получения подписи, все необходимые сведения отправляются по локальной сети Клиенту СРП.

Когда Клиент СРП сообщает результат выполнения команды, функция вызывает `onDone` или `onError` без передачи аргументов, в зависимости от результата выполнения команды.

В случае успешного выполнения команды, вызывается функция `onReturn`, которой сообщается два аргумента: результат выполнения команды и сам объект команды.

Файл исходного кода Клиентской библиотеки СРП “`appSRP.js`” находится в репозитории в папке `ClientLib SRP`.

### 3.3. Разработка набора веб-приложений, использующих программное обеспечение для расширения полномочий доверенных веб-страниц

Для тестирования эффективности и демонстрации возможностей разработанного ПО для расширения полномочий доверенных веб-страниц, был разработан набор веб-приложений, которые используют СРП. Данные приложения должны были удовлетворять следующим требованиям:

- Приложение должно запускаться в обычном браузере;
- Сервер, на котором размещено приложение, должен подписывать хеш команды;
- Приложение должно быть зарегистрировано на Портале СРП;
- Приложение должно иметь ссылку на информацию о приложении на Портале СРП;
- Приложение не должно завершаться аварийно;

В качестве минимального набора демонстрационных приложений, были разработаны следующие веб-приложения:

- Текстовый редактор (далее – Блокнот): данное приложение позволило протестировать выполнение команд на чтение файла и на запись файла
- Файловый менеджер (далее – Проводник): данное приложение позволило протестировать выполнение команд на получение списка файлов в папке, удаление файлов и папок, а также на создание папок и переименование;

В соответствии с вышеуказанными требованиями, данные приложения были разработаны, используя аналогичный технологический стек: для серверной части использовались язык Python и фреймворк Flask, а также Серверная библиотека СРП. Для реализации клиентской части веб-приложений, то есть веб-страниц, использовались HTML, JavaScript, CSS, Bootstrap и Клиентская библиотека СРП.

Оба приложения были зарегистрированы на Портале СРП и были одобрены.

Интерфейс приложения Блокнот содержит в себе минимальный набор элементов интерфейса, достаточный для выполнения прикладной задачи – редактирование текстовых файлов по известному их пути (рисунок 19).

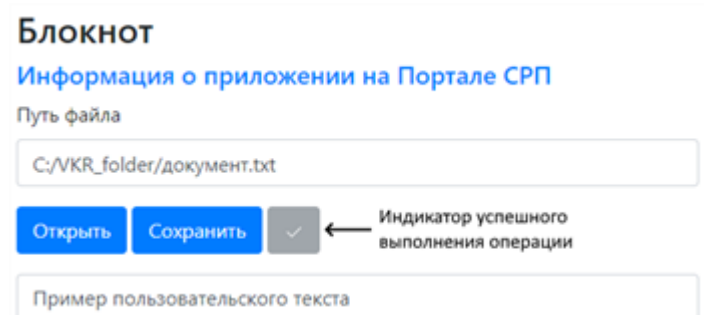


Рис. 18. – Интерфейс приложения Блокнот с примером текста и пути файла

Приложение Блокнот позволяет пользователю открывать текстовые файлы, сообщая приложению их путь, вносить в них изменения и сохранять, а также создавать новые файлы, вписывая их путь и текст в текстовое поле. Индикатор, обозначенный на рисунке 18, позволяет пользователю видеть, в какой стадии находится выполнение команды: символ “галочка” означает, что команда была выполнена успешно, символ “крестик” – что неуспешно, а символ

“песочные часы” – что команда ещё выполняется. Клиентский JavaScript-код Блокнота осуществляет преобразование текста в список байтов при записи файла и списка байтов в текст при чтении файла.

Файлы исходного кода Блокнота расположены в репозитории в папке Notepad:

- Notepad.ipynb – файл с исходным кодом сервера веб-приложения;
- appServerSRP.py – файл с Серверной библиотекой СРП;
- templates – папка с исходным кодом клиента веб-приложения:
  - appSRP.js – файл с Клиентской библиотекой СРП;
  - notepad.html – файл с исходным кодом веб-страницы Блокнота;

Используя тот же набор технологий, было разработано приложение Проводник. Интерфейс Проводника позволяет пользователю просматривать папки файловой системы и осуществлять в них операции удаления, переименования и создания папок (рисунок 19).

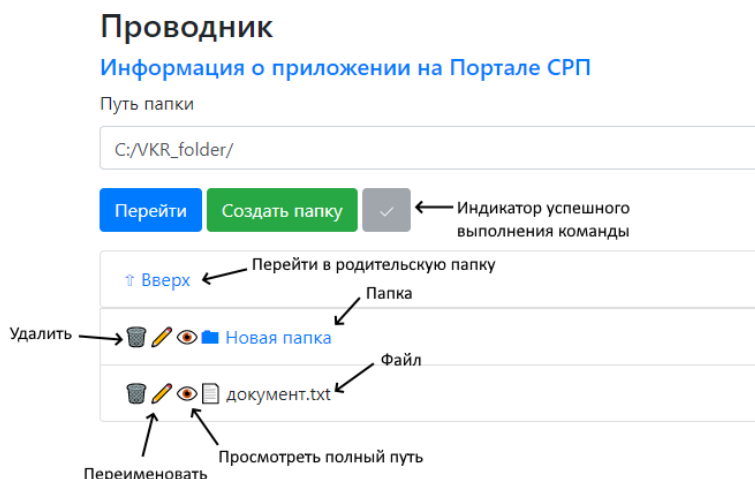


Рис. 19 – интерфейс Проводника с примером пользовательских файлов

Соответственно, используя Проводник, пользователь может также переходить в родительскую папку, при наличии таковой, и запрашивать полный путь файлов и папок. Благодаря получению полного пути, пользователь может скопировать полный путь в соответствующее поле Блокнота и открыть, таким образом, текстовый файл, увиденный в папке с помощью Проводника.

Файлы исходного кода Проводника расположены в репозитории в папке Explorer:

- Explorer.ipynb – файл с исходным кодом сервера веб-приложения;
- appServerSRP.py – файл с Серверной библиотекой СРП;
- templates – папка с исходным кодом клиента веб-приложения:
  - appSRP.js – файл с Клиентской библиотекой СРП;
  - explorer.html – файл с исходным кодом веб-страницы Блокнота;

## ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы было разработано ПО для расширения полномочий доверенных веб-страниц, состоящее из программных компонентов и библиотек, которые позволяют пользователям использовать веб-приложения с расширенными полномочиями, создаваемые независимыми разработчиками.

Система одобрения приложений администрацией позволяет использовать данное ПО в коммерческих целях, особенно, при расширении функционала, так как одобрение может происходить после оплаты разработчиком временной лицензии и подписания договора с администрацией, что положительно скажется на безопасности пользователей.

Исходный код и файлы данной работы были выложены в репозиторий на GitHub<sup>14</sup>.

Данная работа выполнена самостоятельно

«\_\_» \_\_\_\_\_ 20\_\_ г. \_\_\_\_\_ (подпись) \_\_\_\_\_ (подпись)

---

<sup>14</sup> Репозиторий на СРП на GitHub [Интернет-ресурс]  
URL: [https://github.com/MaximNosko/VKR\\_SRP](https://github.com/MaximNosko/VKR_SRP) (дата обращения 06.05.2021)

## СПИСОК ЛИТЕРАТУРЫ

1. Статистика продажи ноутбуков за 2020 год [Интернет-ресурс] URL: [https://www.cnews.ru/news/line/2021-01-20\\_v\\_2020\\_godu\\_rossiyane\\_priobreli](https://www.cnews.ru/news/line/2021-01-20_v_2020_godu_rossiyane_priobreli) (дата обращения 06.05.2021)
2. Характеристики Huawei MateBook D 14 Nbl-WAQ9R [Интернет-ресурс] URL: <https://www.dns-shop.ru/product/542dd7193d9c1b80/14-ultrabuk-huawei-matebook-d-14-nbl-waq9r-seryj/characteristics/> (дата обращения 06.05.2021)
3. Характеристики Huawei MateBook D 15 Nbl-WAQ9R [Интернет-ресурс] URL: <https://www.citilink.ru/product/noutbuk-huawei-matebook-d-boh-waq9r-ryzen-5-3500u-8gb-ssd256gb-vega-8-1372220/> (дата обращения 06.05.2021)
4. Характеристики Honor MagicBook 14 Nbl-WAQ9HNR [Интернет-ресурс] URL: <https://www.dns-shop.ru/product/96a92f403b483332/14-ultrabuk-honor-magicbook-14-nbl-waq9hnr-seryj/> (дата обращения 06.05.2021)
5. Характеристики Lenovo IdeaPad S145-15AST [Интернет-ресурс] URL: <https://www.mvideo.ru/products/noutbuk-lenovo-ideapad-s145-15ast-81n300m7ru-30051486/specification> (дата обращения 06.05.2021)
6. Характеристики Apple MacBook Air 13 i5 1.8/8Gb/128SSD [Интернет-ресурс] URL: <https://www.mvideo.ru/products/noutbuk-apple-macbook-air-13-i5-1-8-8gb-128ssd-mqd32ru-a-30028577> (дата обращения 06.05.2021)
7. Характеристики AMD Ryzen 5 3500U [Интернет-ресурс] URL: <https://www.amd.com/ru/products/apu/amd-ryzen-5-3500u> (дата обращения 06.05.2021)
8. Характеристики AMD A6-9225 [Интернет-ресурс] URL: <https://technical.city/ru/cpu/A6-9225> (дата обращения 06.05.2021)
9. Характеристики Intel Core i5-5350U [Интернет-ресурс] URL: <https://ark.intel.com/content/www/ru/ru/ark/products/84990/intel-core-i5-5350u-processor-3m-cache-up-to-2-90-ghz.html> (дата обращения 06.05.2021)

10. Статистика популярности смартфонов в 2020 году [Интернет-ресурс] URL: <https://www.ixbt.com/news/2020/12/22/xiaomi-5-2020.html> (дата обращения 06.05.2021)
11. Характеристики Samsung Galaxy A51 64GB [Интернет-ресурс] URL: <https://www.mvideo.ru/products/smartfon-samsung-galaxy-a51-64gb-black-sm-a515f-30047894/specification> (дата обращения 06.05.2021)
12. Характеристики Samsung Galaxy A10 [Интернет-ресурс] URL: <https://www.mvideo.ru/products/smartfon-samsung-galaxy-a10-2019-32gb-black-sm-a105f-30043603/specification> (дата обращения 06.05.2021)
13. Характеристики Samsung Galaxy A01 [Интернет-ресурс] URL: <https://www.dns-shop.ru/product/a185ca4535c23332/57-smartfon-samsung-galaxy-a01-16-gb-chnyj/characteristics/> (дата обращения 06.05.2021)
14. Характеристики Apple iPhone [Интернет-ресурс] URL: <https://www.citilink.ru/product/smartfon-apple-iphone-11-64gb-mwlv2ru-a-krasnyi-1178776/properties/> (дата обращения 06.05.2021)
15. Характеристики APPLE iPhone 11 64Gb [Интернет-ресурс] URL: <https://www.citilink.ru/product/smartfon-samsung-galaxy-a51-128gb-sm-a515f-chnyi-1207767/properties/> (дата обращения 06.05.2021)
16. Характеристики Samsung Exynos 9611 [Интернет-ресурс] URL: <https://www.samsung.com/semiconductor/minisite/exynos/products/mobileprocessor/exynos-9611/> (дата обращения 06.05.2021)
17. Характеристики Samsung Exynos 9611 [Интернет-ресурс] URL: <https://en.wikichip.org/wiki/samsung/exynos/9611> (дата обращения 06.05.2021)
18. Характеристики Samsung Exynos 7884B [Интернет-ресурс] URL: <https://www.notebookcheck.net/Samsung-Exynos-7884B-SoC-Benchmarks-and-Specs.427997.0.html> (дата обращения 06.05.2021)



19. Характеристики Qualcomm Snapdragon 439 [Интернет-ресурс] URL: <https://rankquality.com/qualcomm-snapdragon-439/> (дата обращения 06.05.2021)
20. Характеристики Характеристики APPLE iPhone 11 64Gb [Интернет-ресурс] URL: <https://www.svyaznoy.ru/catalog/phone/224/5633084/specs> (дата обращения 06.05.2021)
21. Характеристики [Интернет-ресурс] URL: <https://en.wikichip.org/wiki/apple/ax/a13> (дата обращения 06.05.2021)
22. Характеристики [Интернет-ресурс] URL: <https://www.mvideo.ru/products/smartfon-samsung-galaxy-a51-128gb-black-sm-a515f-30047891/specification> (дата обращения 06.05.2021)
23. Характеристики Samsung Exynos 9611 [Интернет-ресурс] URL: <https://en.wikichip.org/wiki/samsung/exynos/9611> (дата обращения 06.05.2021)
24. Характеристики iPhone 11 [Интернет-ресурс] URL: <https://www.ixbt.com/news/2019/09/13/4-iphone-11-iphone-11-pro-iphone-11-pro-max-antutu.html> (дата обращения 06.05.2021)
25. Самые посещаемые сайты рунета в 2020 году [Интернет-ресурс] URL: <https://infoselection.ru/infokatalog/internet-i-programmy/internet-osnovnoe/item/90-50-samykh-poseshchaemykh-sajtov-runeta> (дата обращения 06.05.2021)
26. Самые популярные мобильные приложения в 2020 году [Интернет-ресурс] URL: [https://yamobi.ru/posts/opublikovan\\_reyting\\_populyarnosti\\_mobilnyih\\_prilozheniy\\_2020\\_goda.html](https://yamobi.ru/posts/opublikovan_reyting_populyarnosti_mobilnyih_prilozheniy_2020_goda.html) (дата обращения 06.05.2021)
27. Запуск “Электронной сберкассы” [Интернет-ресурс] URL: <https://www.banki.ru/news/daytheme/?id=561620> (дата обращения 06.05.2021)

- 28.Выход приложения Сбербанка [Интернет-ресурс] URL: <http://futurebanking.ru/post/3100> (дата обращения 06.05.2021)
- 29.Запуск Авито [Интернет-ресурс] URL: <https://trends.rbc.ru/trends/sharing/cmrm/5fe0b3169a794799f50e1bd2> (дата обращения 06.05.2021)
- 30.Выход приложения Авито[Интернет-ресурс] URL: <https://www.content-review.com/articles/17879/> (дата обращения 06.05.2021)
- 31.Запуск ВКонтакте[Интернет-ресурс] URL: <https://vk.com/blog?nid=152> (дата обращения 06.05.2021)
- 32.Выход приложения ВКонтакте[Интернет-ресурс] URL: <https://habr.com/ru/post/121854/> (дата обращения 06.05.2021)
- 33.Запуск YouTube [Интернет-ресурс] URL: <https://www.britannica.com/topic/YouTube> (дата обращения 06.05.2021)
- 34.Выход приложения YouTube[Интернет-ресурс] URL: <https://techcrunch.com/2010/07/07/youtube-iphone-mobile-html5/> (дата обращения 06.05.2021)
- 35.Запуск AliExpress [Интернет-ресурс] URL: <https://www.alibabagroup.com/en/about/businesses> (дата обращения 06.05.2021)
- 36.Выход приложения AliExpress[Интернет-ресурс] URL: <https://4pda.ru/forum/index.php?showtopic=405551> (дата обращения 06.05.2021)
- 37.Рейтинг браузеров [Интернет-ресурс] URL: <https://gs.statcounter.com/> (дата обращения 06.05.2021)
- 38.Дата выхода Google Chrome[Интернет-ресурс] URL: <https://googleblog.blogspot.com/2008/09/google-chrome-now-live.html> (дата обращения 06.05.2021)

- 39.Дата выхода Safari [Интернет-ресурс] URL: <https://dot.kde.org/2003/01/08/apple-announces-new-safari-browser> (дата обращения 06.05.2021)
- 40.Дата выхода Mozilla [Интернет-ресурс] URL: <https://www.mozilla.org/ru/about/history/> (дата обращения 06.05.2021)
- 41.Геолокация в Chrome [Интернет-ресурс] URL: <https://chrome.softwaredownload.co.in/chrome-5-0-375> (дата обращения 06.05.2021)
- 42.Геолокация в Safari [Интернет-ресурс] URL: [https://www.researchgate.net/figure/Safari-5-requests-the-users-permission-before-granting-a-website-access-to-geolocation\\_fig1\\_228960320](https://www.researchgate.net/figure/Safari-5-requests-the-users-permission-before-granting-a-website-access-to-geolocation_fig1_228960320) (дата обращения 06.05.2021)
- 43.Геолокация в Mozilla [Интернет-ресурс] URL: <https://hacks.mozilla.org/2009/06/geolocation/> (дата обращения 06.05.2021)
- 44.Уведомления в браузерах [Интернет-ресурс] URL: [https://caniuse.com/mdn-api\\_notification](https://caniuse.com/mdn-api_notification) (дата обращения 06.05.2021)
- 45.Микрофон и камера в Chrome [Интернет-ресурс] URL: <https://chrome.softwaredownload.co.in/chrome-21-0-1180> (дата обращения 06.05.2021)
- 46.Микрофон и камера в Safari [Интернет-ресурс] URL: <https://habr.com/ru/company/flashphoner/blog/339396/> (дата обращения 06.05.2021)
- 47.Микрофон и камера в Mozilla [Интернет-ресурс] URL: [https://website-archive.mozilla.org/www.mozilla.org/firefox\\_releasenotes/en-us/firefox/18.0/releasenotes/](https://website-archive.mozilla.org/www.mozilla.org/firefox_releasenotes/en-us/firefox/18.0/releasenotes/) (дата обращения 06.05.2021)
- 48.Руссинович М., Соломон Д., Ионеску А., Йосифович П. В60 Внутреннее устройство Windows. 7-е изд. — СПб.: Питер, 2018.

49. Носко М.М. Создание веб-приложения для работы с файлами Jupyter Notebook в рамках учебного процесса // Научные записки молодых исследователей – 2018. – №4. – С. 53-59.
50. Англо-русский толковый научно-технический словарь по системному анализу, программированию, электронике и электроприводу: В 2-х т. Т. 1. – Томск, 2008.
51. Росс В. С. Создание сайтов: HTML, CSS, PHP, MySQL. Учебное пособие, ч. 1 — МГДД(Ю)Т, М.:2010 – 107 с.
52. Богданенко Д. А. Подходы к архитектурному проектированию веб – приложений (рус.) // Молодой ученый – 2018. – № 9 (195). – С. 24—29.
53. Прохоренок Н.А. Python 3 и PyQt. Разработка приложений. СПб.; 2012.
54. Lutz M. Python Pocket Reference, Fifth Edition. Sebastopol, CA; 2014.
55. Васильева Т.Н. Криптографическая защита цифровых носителей информации // Техничко-технологические проблемы сервиса – 2019. – №4 – С. 50-54
56. Волков А.С., Волкова К.А. Обзор архитектурных компонентов современного веб-приложения // Аллея науки, т.3 – 2019. – №1. – С. 958-961