Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление подготовки 09.03.02 – Информационные системы и технологии
Направленность (профиль) образовательной программы Безопасность информационных систем

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой
_____ А.В. Бушманов
«_____»_____2019 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка кроссплатформенного клиент-серверного приложения «Менеджер паролей»

Исполнитель
студент группы 555-об _____ А.В. Понизов
(подпись, дата)

Руководитель
доцент, канд.техн. наук _____ Т.А. Галаган
(подпись, дата)

Консультант
по безопасности и экологичности
доцент, канд. техн. наук _____ А.Б. Булгаков
(подпись, дата)

Нормоконтроль
инженер кафедры _____ В.Н. Адаменко
(подпись, дата)

Благовещенск 2019

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВПО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ
Зав. кафедрой
_____ А.В. Бушманов
«____» _____

З А Д А Н И Е

К выпускной квалификационной работе студента Понизова Александра Викторовича
1. Тема дипломной работы: Разработка кроссплатформенного клиент-серверного приложения «Менеджер паролей»
(утверждена приказом от 15.04.2019 №847-уч)
2. Срок сдачи студентом законченной работы: 25.06.2019 г.
3. Исходные данные к выпускной квалификационной работе: отчет о прохождении преддипломной практики, нормативная документация, специальная литература.
4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов): обоснование необходимости разработки и определение требований, проектирование программного продукта, оценка надежности и качества функционирования объекта проектирования, руководство пользователя, описание способов защиты информации для программы, обоснование безопасности и экологичности продукта.
6. Консультанты по выпускной квалификационной работе:
по безопасности и экологичности – Булгаков А.Б., доцент, кандидат технических наук.
7. Дата выдачи задания: 16.04.2019 г.
Руководитель выпускной квалификационной работы: Галаган Т.А., доцент, кандидат технических наук.
Задание принял к исполнению:
_____

РЕФЕРАТ

Выпускная квалификационная работа содержит 80с, 34 рисунка, 14 таблиц, 5 приложений, 19 источников, 7 нормативных ссылок.

РАЗРАБОТКА, АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ, ПРОЕКТИРОВАНИЕ, КЛИЕНТ-СЕРВЕРНАЯ АРХИТЕКТУРА, JWT, ИНФОРМАЦИОННАЯ СИСТЕМА.

В работе произведено проектирование и разработка кроссплатформенного клиент-серверного приложения «Менеджер паролей».

Цель работы: проектирование и разработка приложения «Менеджер паролей»

Выполнение проекта включает пять этапов:

Первым этапом является произведение анализа предметной области. В результате были рассмотрены актуальные угрозы безопасности конфиденциальным данным в современных приложениях, произведен анализ существующих решений и их защищенности, на основании результатов которого обоснована потребность в проектировании и разработки модуля.

На втором этапе выполняется определение целей и функций приложения, спроектирована структура приложения. Также проведена характеристика функциональных модулей системы и проектирование базы данных. Сформированы требования к программному продукту. В результате проектирования получены: характеристика функциональных модулей приложения «Менеджера паролей», требования к программному продукту, и схема IDEF1X базы данных.

На третьем этапе выполняется выбор методологии разработки программного обеспечения, выбор средств разработки и разработка модулей, в результате была получена схема взаимодействия модулей приложения «Менеджер паролей», приложение «Менеджер паролей», а также пример реальной эксплуатации приложения-сервера.

На четвертом этапе для разработанного приложения были изучены модули угроз и варианты их реализации нарушителем, а также меры по их предотвращению.

На пятом этапе установлены рекомендации по безопасности и экологичности.

Результатом выполнения выпускной квалификационной работы является кроссплатформенное клиент-серверное приложение «Менеджер паролей».

СОДЕРЖАНИЕ

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей бакалаврской работе использованы ссылки на следующие стандарты и нормативные документы:

1 ГОСТ Р ИСО/МЭК 27002-2012. Информационная технология (ИТ). Методы и средства обеспечения безопасности. Свод норм и правил менеджмента информационной безопасности.

2 ГОСТ Р 51583-2014 Защита информации. Порядок создания автоматизированных систем в защищенном исполнении. Общие положения.

3 ГОСТ Р ИСО 9241 – 2016. Эргономика взаимодействия человек-система. Часть 161. Элементы графического пользовательского интерфейса.

4 ГОСТ 30772-2001. Межгосударственный стандарт. Ресурсосбережение. Обращение с отходами.

5 СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы

6 СТО СМК 4.2.3.21-2018 Оформление выпускных квалификационных и курсовых работ (проектов)

7 Указ Президента РФ от 06.03.1997 N 188 (ред. от 13.07.2015) «Об утверждении перечня сведений конфиденциального характера».

СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

ЯП – язык программирования;

HTTP – hypertext transfer protocol – протокол передачи гипертекста;

HTTPS – hypertext transfer protocol secure – безопасный протокол передачи гипертекста;

JWT – Json Web Token – токен авторизации;

REST – Representational State Transfer;

API – Application Programming Interface;

DDD – Domain-Driven Design;

Onion Architecture – «Луковичная архитектура».

ВВЕДЕНИЕ

Несмотря на то, что в мире придумано много способов аутентификации и контроля доступа, именно пароль самый распространенный и одновременно наиболее уязвимый. Множество интернет-порталов и сервисов в целях безопасности запрещают пользователям создавать простые пароли, что, с одной стороны, хорошо, а с другой просто неудобно. Если же принять во внимание факт существования десятка подобных сайтов, то сразу становится заметна проблема хранения паролей. Для устранения пробела между человеческим фактором и безопасностью данных на помощь приходят менеджеры паролей, которые берут на себя организацию паролей пользователя. Однако при таком подходе получается, что безопасность пользователя зависит только от мастер-пароля.

К информационным системам, которые включают в себя компонент взаимодействия с конфиденциальными данными, предоставляются определенные требования безопасности. Ответственность по разработке и соблюдению этих требований целиком лежит на команде, проектирующей и разрабатывающей эту ИС. Но, в частности, каждый пользователь информационной системы должен соблюдать определённые правила, для обеспечения сохранности своих данных. Для этого в каждой современной информационной системе при регистрации, пользователя просят создать пароль, или же система генерирует и предоставляет пользователю безопасный временный пароль, который подлежит немедленной принудительной замене после входа в систему (ГОСТ Р ИСО/МЭК 27002-2012). В этом же стандарте предъявляются требования не только к качественной составляющей пароля (длина, использование строчных/заглавных букв, цифр, знаков), но и к методам доставки и смены паролей. Для каждого отдельного сервиса рекомендуется использовать различные пароли, в случаях, когда пароль мог быть скомпрометирован он подлежит немедленной замене. В теории, это обеспечивает необходимый уровень безопасности таких систем, однако, если злоумышленник каким либо образом получит

доступ к сервису от лица пользователя, то он получит доступ и к данным который тот, в свою очередь, может содержать: данные банковских карт, другие пароли, паспортные данные, адрес электронной почты или другую конфиденциальную информацию.

Так же, стоит отметить, что целью злоумышленника часто является не отдельный пользователь, а сервер, на котором происходит обработка конфиденциальной информации. Это означает, что операционная система сервера, базы данных так же должны иметь стойкий пароль. Одна из самых значимых и распространенных проблем “паролей”, это их простота, что несомненно является верным утверждением, однако, следуя из вышесказанного, с уверенностью можно сказать, что проблема может заключаться в неправильном менеджменте паролей как пользователем, так и системным администратором. В настоящее время данные проблемы являются актуальными и их необходимо решать.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Анализ безопасности конфиденциальных данных в системах типа “Менеджер Паролей”

Объектом исследования выпускной квалификационной работы является проектирование и разработка кроссплатформенного клиент-серверного приложения «Менеджер паролей». Предметом исследования – исследование безопасности существующих решений «Менеджер паролей». Результатом – разработанное кроссплатформенное приложение, предоставляющего пользователю безопасную и удобную среду для управления личными данными (Менеджер паролей).

Конфиденциальная информация – обязательное для выполнения лицом, получившим доступ к определенной информации, требование не передавать такую информацию третьим лицам без согласия ее обладателя. Согласно Закону РФ «Об информации», информация – сведения (сообщения, данные) независимо от формы их представления.

Согласно Указ Президента РФ от 06.03.1997 N 188 (ред. от 13.07.2015) «Об утверждении перечня сведений конфиденциального характера», к конфиденциальной информации относятся:

- Сведения о фактах, событиях и обстоятельствах частной жизни гражданина, позволяющие идентифицировать его личность (персональные данные), за исключением сведений, подлежащих распространению в средствах массовой информации в установленных федеральными законами случаях.
- Сведения, составляющие тайну следствия и судопроизводства, а также сведения о защищаемых лицах и мерах государственной защиты.
- Служебные сведения, доступ к которым ограничен органами государственной власти в соответствии с Гражданским кодексом Российской Федерации и федеральными законами (служебная тайна).
- Сведения, связанные с профессиональной деятельностью, доступ к которым ограничен в соответствии с Конституцией Российской Федерации и

федеральными законами (врачебная, нотариальная, адвокатская тайна, тайна переписки, телефонных переговоров, почтовых отправлений, телеграфных или иных сообщений и так далее).

- Сведения, связанные с коммерческой деятельностью, доступ к которым ограничен в соответствии с Гражданским кодексом Российской Федерации и федеральными законами (коммерческая тайна).
- Сведения о сущности изобретения, полезной модели или промышленного образца до официальной публикации информации о них.

Порядок создания автоматизированных информационных систем в защищенном исполнении изложен в ГОСТ Р 51583-2014. Данный стандарт содержит общие положения и введен в действие приказом Федерального агентства по техническому регулированию и метрологии от 28 января 2014 г. №3-ст. Он распространяется на создаваемые информационные системы, на которые заказчиком или же законодательством наложены требования по защите:

- **мероприятия по защите информации** – совокупность действий, направленных на реализацию способов или средств защиты информации;
- **обработка информации** – выполнение любого действия или их совокупности над информацией;
- **система защиты информации информационной системы –** совокупность средств контроля эффективности защиты и организационных мероприятий, технических, программных и программно-технических средств защиты информации;
- **информационная система –** технологические средства и технологии предназначенные для обработки информации, хранящейся в различных видах хранилищ**.**

1.2 Актуальные угрозы безопасности конфиденциальным данным в современных приложениях

При разработке приложений следует учитывать, что данные, которые оно использует могут предоставлять определенный интерес для третьих лиц.

Степень ценности этих данных может варьироваться в огромных пределах, тем не менее, даже наиболее простую приватную информацию – пароль для входа в приложение, необходимо защищать. Особенно это актуально в приложениях, используемых в сфере финансовых, банковских операций и передачи, хранения информации.

Перечень актуальных атак на мобильное приложение:

а) Декомпиляция файла приложения и извлечение данных из локальных хранилищ. При создании приложений с использованием сред аналогичных Java или .Net, злоумышленник имеет возможность декомпилирования приложений в исходный код, с получением оригинала исходного кода, вплоть до комментариев разработчиков.

б) Перехват данных, передаваемых по сети (MITM – атаки) – атака, в которой злоумышленник тайно ретранслирует или каким-либо способом изменяет связь между двумя сторонами, в то время как эти стороны считают, что общаются друг с другом.

в) Атаки на устройства под правами администратора – разновидность атак, в которых, вредоносное программное обеспечение или злоумышленник, действуя на устройстве пользователя и используя права администратора, выполняет различные действия.

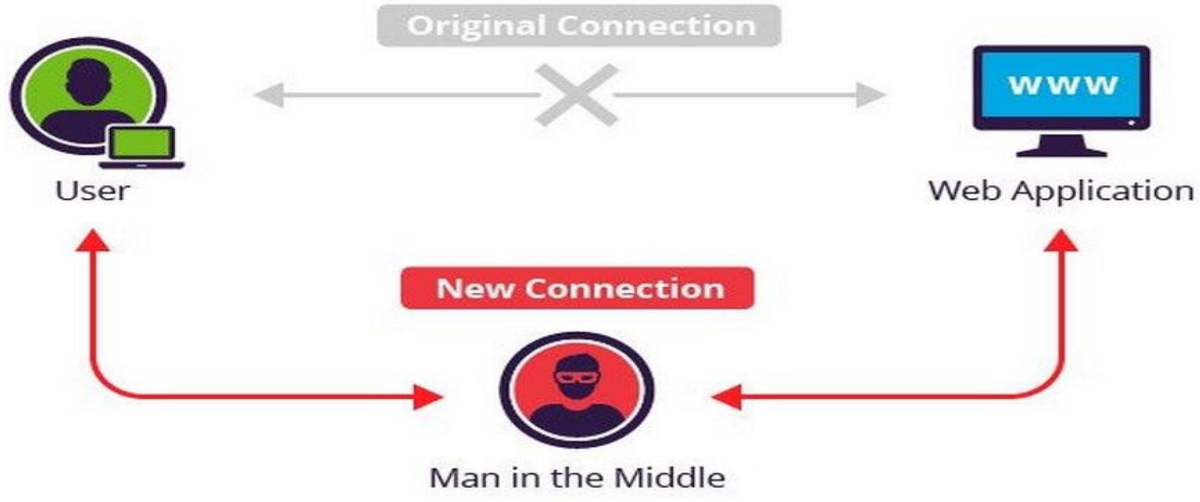

Рисунок 1 – Атака посредника (MITM – атака)

Для определения вектора атак, рассмотрим основные виды уязвимостей:

а) Хранение критически важной информации в коде. Разработчик хранит КВИ внутри кода (в константах, ресурсах и т. п.). Например: хранение ключей шифрования в виде открытой строки, хранение соли для создания хеш-сумм или паролей, и т. п. Данная уязвимость используется в атаке, включающей в себя декомпиляцию файла приложения, после чего злоумышленник получит доступ к КВИ.

б) Использование криптографических алгоритмов с хранением ключей. Ключи вынужденно сохраняются в базе данных или другом локальном хранилище без должного уровня защиты.

в) Использование любого ассиметричного алгоритма шифрования с приватным ключом, известным серверу. В этом случае, если сервер будет взломан злоумышленником, то атакующий получит доступ к личным данным пользователя. Это нарушает конфиденциальность данных пользователя, ведь его данные в любой момент времени могут быть расшифрованы. Серверная часть защищенных приложений не должна содержать средств для расшифровки данных пользователя без их ведома.

г) Обмен открытыми данными между клиентом, сервером и хранилищем – доступ к данным будет получен в случае атаки типа MITM, или получения атакующем доступа к не защищенному хранилищу.

д) Использование самописных или не валидных алгоритмов защиты – любое отклонение от существующих, многократно проверенных и математически доказанных алгоритмов шифрования, приводит к почти полной ликвидации безопасности в приложении.

е) Все из перечисленных уязвимостей так или иначе существуют в современных приложениях, однако при разработке защищенного приложения, они должны быть учтены. Существуют определенные методы и средства, призванные помочь разработчику обеспечить необходимый уровень безопасности в приложении.

1.3 Методы и средства обеспечения безопасности в современных приложениях

Существует несколько общих правил в разработке защищенных приложений, которые необходимо соблюдать для обеспечения должного уровня безопасности:

- Защита приложения аутентификацией и идентификацией – механизм защиты, когда приложение, защищенное паролем, запрашивает его при каждой попытке входа для аутентификации и идентификации пользователя. В том случает если приложение работает в фоновом режиме, то, при переходе в активный режим, оно должно запросить защитный код (пароль), перекрывая при этом экран приложения.
- Ограничение попыток неверного ввода пароля – ограничение, при котором, в случае многократного неверного ввода пароля, сеанс пользователя с приложением, если таковой имеется, должен быть прекращен, а пароль пользователя должен быть признан скомпрометирован с возможностью его последующей замены, по защищенным каналам.
- Приложение не должно отображать конфиденциальную информацию большими, яркими, хорошо читаемыми при любых условиях шрифтами, без явной на то необходимости или без запроса пользователя.
- Исключить непроверенные библиотеки с открытым кодом, которые предлагают некие средства защиты пользовательских данных.
- Недопустимо использование криптографических библиотек с закрытым исходным кодом – в таких решениях невозможно проверить качество предоставляемых функций и их соответствие стандартам.
- В публичной версии приложения логгирование должно быть либо отключено, либо логи должны быть зашифрованы.

Так же существуют определенные рекомендации для клиент-серверных приложений:

- Использование сессионных механизмов с ограниченным временем жизни сессии позволяет исключить простаивание приложения в незащищенном режиме, когда пользователь забыл выйти/закрыть приложение.
- Клиент-серверное приложение не должно производить изменение личных данных пользователя в локальном режиме, каждое изменение должно быть синхронизировано с сервером.
- Использование клиентом абсолютного времени сервера позволяет исключить факторы, связанные с махинациями со временем на стороне клиента.

В случае если время абсолютно и длительность жизни сессии ограничена сервером, то в независимости от того, как изменится время клиента, сессия будет закрыта в нужный момент.

Соблюдение данных правил и рекомендаций способно повысить качество и надежность приложения, а также исключить появление скрытых уязвимостей. В пункте 1.2 рассматривался список возможных атак, а также уязвимостей, которые эти атаки используют.

Для устранения таких уязвимостей необходимо соблюдать следующие рекомендации:

а) Разработчики должны избегать хранения критически важной информации в коде, такой как ключи шифрования, «соль», используемая для генерации хеш-сумм. Данную информацию необходимо перенести либо в защищенное хранилище, либо генерировать её каждый раз при необходимости.

б) В случае использования ассиметричных алгоритмов шифрования для обмена данными между клиентом и сервером, эти данные не должно быть возможно окончательно расшифровать на стороне сервера. Это достигается, как правило, использованием симметричного шифрования на стороне клиента зависящего от ключа пользователя, с последующей передачей по защищенному каналу.

в) Следует подбирать подходящий алгоритм только из отлаженных и актуальных общеизвестных криптографических алгоритмов.

1.4 Сравнительный анализ существующих решений типа «Менеджер Паролей»

“Менеджер паролей” – это программное обеспечение, которое обеспечивает хранение пользовательских паролей в зашифрованном виде на устройстве пользователя или в доверенном хранилище. “Менеджер паролей” исключает необходимость запоминания множества паролей, что позволяет создавать уникальный пароль для каждого сервиса, так же длина и сложность пароля становится не ограниченной.

Исходя из назначения, “менеджер паролей” должен обладать следующими свойствами:

- Защищенность – данные пользователя, должны быть защищены как от программных атак, так и от кражи извне.
- Отказоустойчивость – данные пользователя должны быть доступны пользователю, если не для чтения, то для восстановления в последующем.
- Универсальность и кроссплатформенность – пользователь должен иметь возможности получить, сохранить, просмотреть и передать свои данные на любое доверенное устройство.
- Совместимость версий – все версии программного обеспечения должны быть совместимы между собой с целью создания комфортной среды для пользователя. Не должна возникать ситуация, когда пользователь может взаимодействовать со своими данными в новой версии, но в более старой версии поддержка работы с этими данными не реализована даже частично.

В качестве испытуемых было выбрано три наиболее популярных решений для ОС Windows:

- Kaspersky Password Manager 5.0.0.176;
- Sticky Password 7.0.2.27;
- 1Password 1.0.9.337.

Каждый из них нужно проверить на уязвимость к следующим атакам:

- атака на мастер-пароль – атака, которая всего реализуется посредством кейлоггеров или посредством создания снимков экрана. Однако в рамках

данного исследования используется эксплойт использующий функцию Win32 API SendMessage с параметром WM_GETTEXT. Данный вариант не вызывает реакции со стороны антивирусов;

- атака на содержимое базы паролей – используется тот же метод, что и при атаке на мастер-пароль;

- атака DLL Hijacking – заключается в подмене динамически загружаемых библиотек на стороннюю, содержащую вредоносный код;

Параметрами для анализа будут служить: устойчивость к атаке и количество похищенных данных.

Результаты анализа занесены в таблицы 1-3.

Kaspersky Password Manager 5

KPM 5 поддается взлому с помощью send_message атаки. Представленных ниже рисунках 2-3 видно, что пароль «This is secret password» был перехвачен.

Также KMP подвержен третьей атаке – DLL Hijacking. При старте приложение выполняет загрузку библиотек bthprops.cpl и cryptsp.dll. В результате подмены появляется выполнить код метода DllEntryPoint который исполнится при инициализации библиотеки.

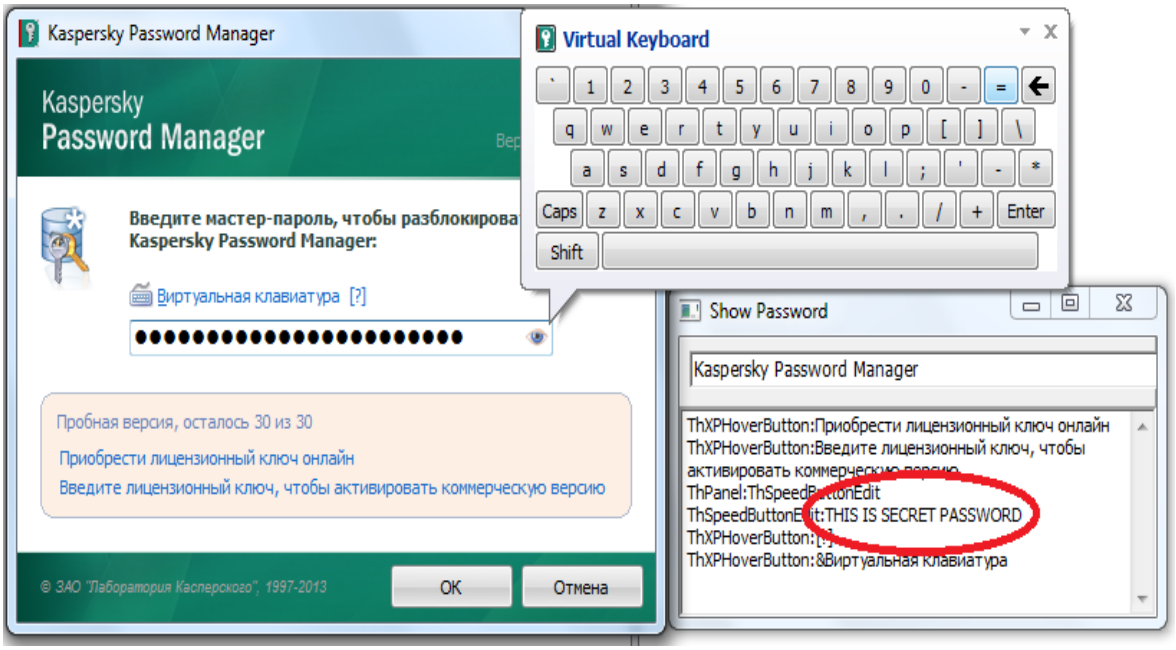

Рисунок 2 – Реализация эксплойта в программе KPM

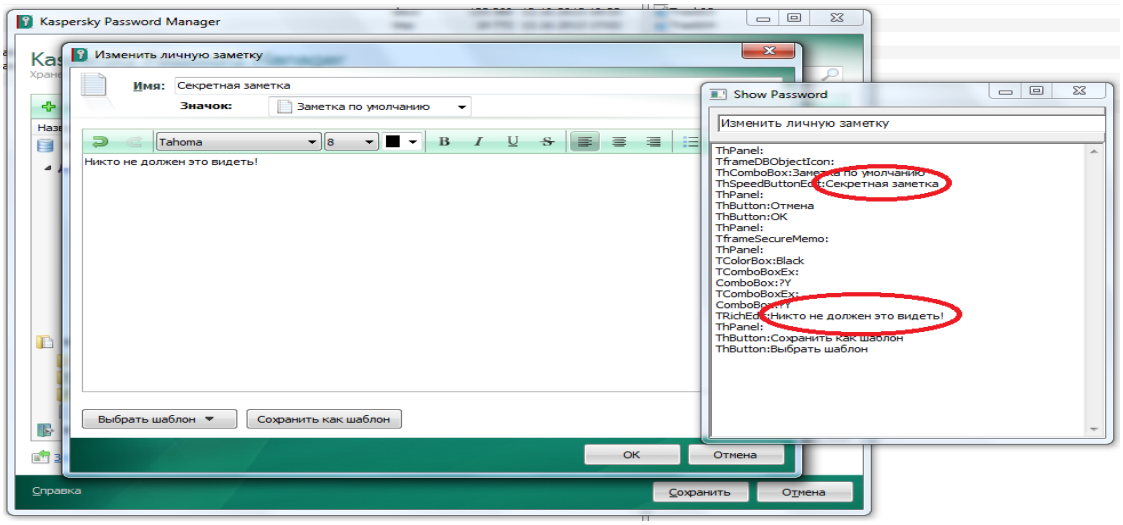

Рисунок 3 – Реализация эксплойта в хранилище паролей

Sticky Password 7.0.2.27

Sticky Password является прародителем Kaspersky Password Manager. Именно на его основе «Лаборатория Касперского» в свое время разработала собственный продукт. Оба менеджера обладают схожим функционалом и структурой, фактически главное их отличие – это номера версий и дизайн. Несмотря на то, что последняя версия Sticky Password вышла совсем недавно, программа по-прежнему не избавилась от всех тех проблем, которыми когда-то наградила своего отпрыска. Первая и вторая атака успешно выдала мастер-пароль и частично содержимое базы паролей.

Такой же результат, как и KPM получен при входе в приложение. Одна уязвимыми библиотеки оказались другие – itlib.dll, olepro32.dll, profapi.dll (библиотеки приложения)

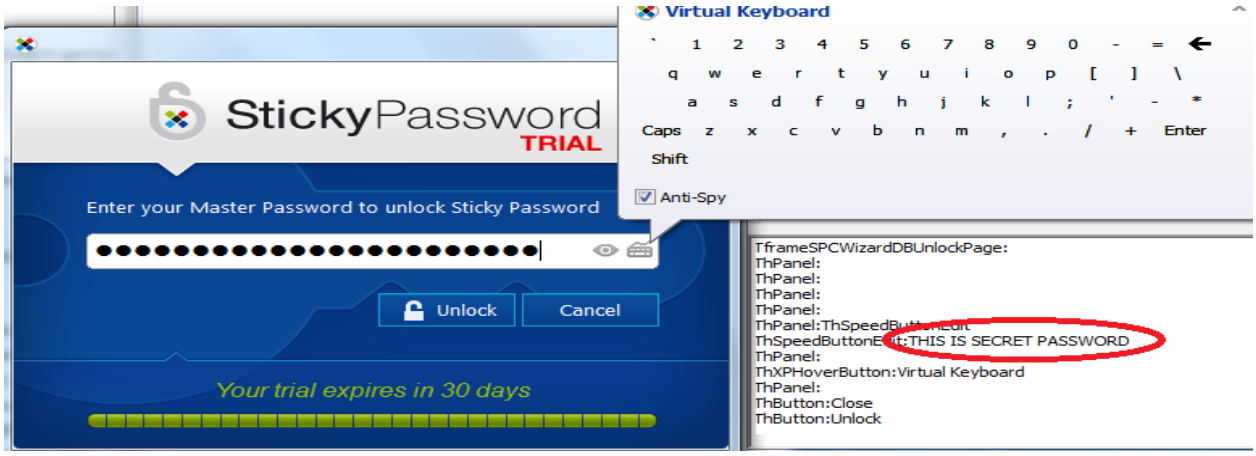

Рисунок 4 – Реализация эксплойта в программе StickyPassword

1Password 1.0.9.337

Не получилось получить мастер-пароль, однако после входа удалось получить все данные, содержащиеся в форме.

Результаты анализа существующих решений приведены в таблицах 1-3.

Таблица 1 – Результаты анализа защищенности существующих решений от атак на мастер-пароль

Наименование продукта	Устойчивость к атаке	Процент похищенных данных
Kaspersky Password Manager 5.0.0.176	Нет	100%
Sticky Password 7.0.2.27	Нет	100%
1Password 1.0.9.337	Частично	Всё кроме мастер-пароля

Таблица 2 – Результаты анализа защищенности хранилища существующих решений

Наименование продукта	Устойчивость к атаке	Процент похищенных данных
Kaspersky Password Manager 5.0.0.176	Частично	Данные удается перехватить в момент просмотра и редактирования
Sticky Password 7.0.2.27	Частично	Данные удается перехватить в момент просмотра и редактирования
1Password 1.0.9.337	Частично	Данные удается перехватить в момент просмотра и редактирования

Таблица 3 – Результаты анализа защищенности существующих решений к атаке DLL Hijacking

Наименование продукта	Устойчивость к атаке	Комментарий
Kaspersky Password Manager 5.0.0.176	Нет	Возможна подмена двух библиотек: bthprops.cpl и cryptsp.dll
Sticky Password 7.0.2.27	Нет	Возможна подмена: Fitlib.dll, olepro32.dll, profapi.dll
1Password 1.0.9.337	Нет	Midimap.dll

Из этого можно сделать вывод, что в настоящее время, несмотря на существование множества «менеджеров-паролей», лишь единицы из них могут обеспечить приемлемый уровень защиты. Для защиты данных пользователя,

необходимо рассмотреть варианты, при которых за авторизацию будет отвечать не локальное приложение, которое поддается воздействию различных эксплойтов, а надежный сервер, доступ к которому будет ограничен. Так же это решение позволит не допустить кражу данных из локального хранилища. Одним из вариантов таких WEB приложений это REST API приложение, состоящее из клиента и REST API сервера. Сервер выполняет роль хранилища информации с функциями предварительной обработки данных.

2 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ «МЕНЕДЖЕР ПАРОЛЕЙ»

2.1 Цели и функции приложения

Назначение приложения, как информационной системы – автоматизировать, структурировать и журнализировать поток конфиденциальной информации типа логин/пароль как для предприятия, так и для конечного пользователя.

В данный момент на предприятиях разного масштаба процессы выдачи и сопровождения аутентификационных пар, как правило, отсутствуют полностью. Системный администратор имеет возможность создавать такие пары в неограниченном количестве. Попытка пользователя восстановить существующий пароль часто сводится к его пересозданию. Хранение и выдача подобных пар, как новым сотрудникам, так и существующим, происходит посредством бумажных носителей. Что приводит к *появлению новых путей реализации угроз информационной безопасности предприятия*.

Основные функции приложения:

- регистрация и авторизация пользователей приложения;
- выход пользователя и удаление учетной записи;
- создание группы (папки) для хранения аккаунтов;
- добавление/удаление/изменение аутентификационных пар;
- возможность поделиться аутентификационной парой;
- поиск внутри группы (папки);
- загрузка данных пользователя на основании авторизации на любое устройство-клиент;
- функция смена темы;

2.2 Структура приложения «Менеджер паролей»

«Менеджер паролей» является кроссплатформенным клиент-серверным приложением.

Функциональная модель программы представлена на рисунке 5.

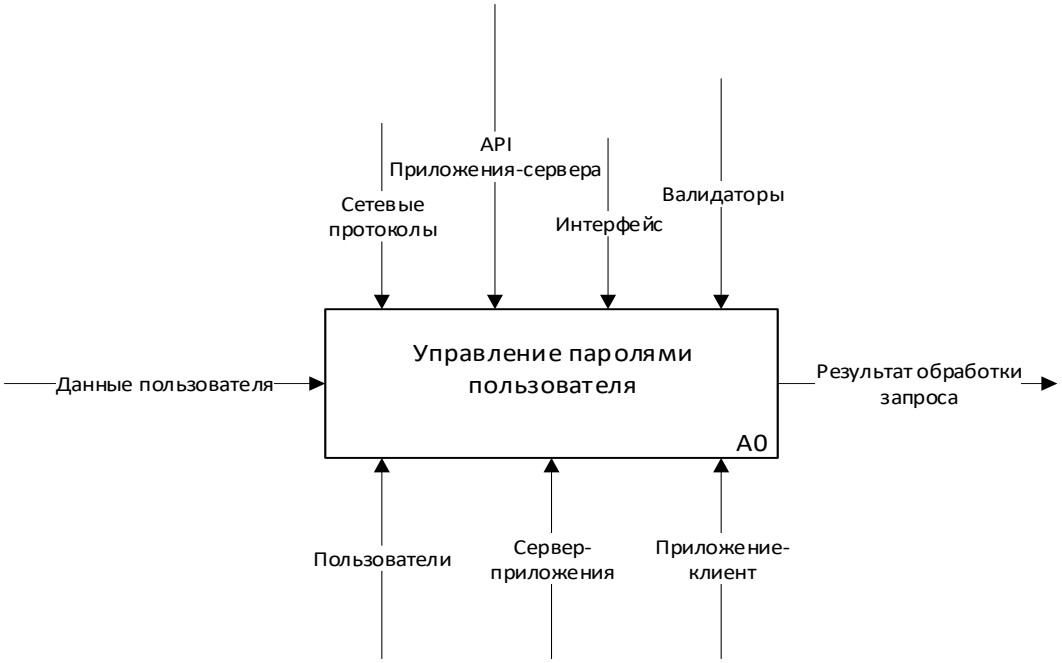

Рисунок 5 – Функциональная модель программы в нотации IDEF0

Рисунок 6 – Декомпозиция функциональной модели приложения

Для начала работы пользователю необходимо зарегистрироваться в приложении клиенте, после чего авторизоваться. Если пользователь ранее был зарегистрирован, то ему необходимо ввести в приложение-клиент необходимые данные. После авторизации пользователю предоставляется весь функционал приложения.

Декомпозиция логики приложения-клиента и логики приложения сервера представлены в Приложении Г (рисунок Г1 и рисунок Г2)

На основании перечисленных выше функций можно выделить следующие модули, формирующие структуру информационной системы:

а) Сервер:

1) Модуль авторизации – производит действия по приему, проверке и выдачи JWT токенов. Так же включает в себя функционал регистрации и удаления аккаунтов пользователя.

2) Модуль REST API – реализует основной функционал приложения, основываясь на REST API, это позволяет приложениям, построенным для разных платформ одинаково взаимодействовать с сервером

3) БД – одна из СУБД (MySQL, MSSQL) для хранения данных клиента

б) Клиент:

1) Модуль интерфейса пользователя

2) Модуль авторизации – логика авторизации, находящаяся на стороне клиента. Основным функционалам является прием данных от пользователя и последующее преобразование их в запрос к серверу. Модуль также поддерживает ожидание ответа от сервера об успехе или не успехе авторизации

3) Модуль клиент-серверной логики – реализует основной функционал приложения: добавление, удаление, редактирование пользовательских данных. Реализует функционал «Поделиться» посредством передачи данных серверу.

4) Модуль обработки данных сессии – занимается обработкой данных текущей сессии пользователя. Предоставляет доступ к данным авторизованного пользователя и его JWT токена.

2.3 Характеристика функциональных модулей системы

2.3.1 Модуль авторизации

Авторизация (англ. authorization «разрешение; уполномочивание») –

процесс предоставления определённому лицу или группе лиц прав на выполнение действий; а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий.

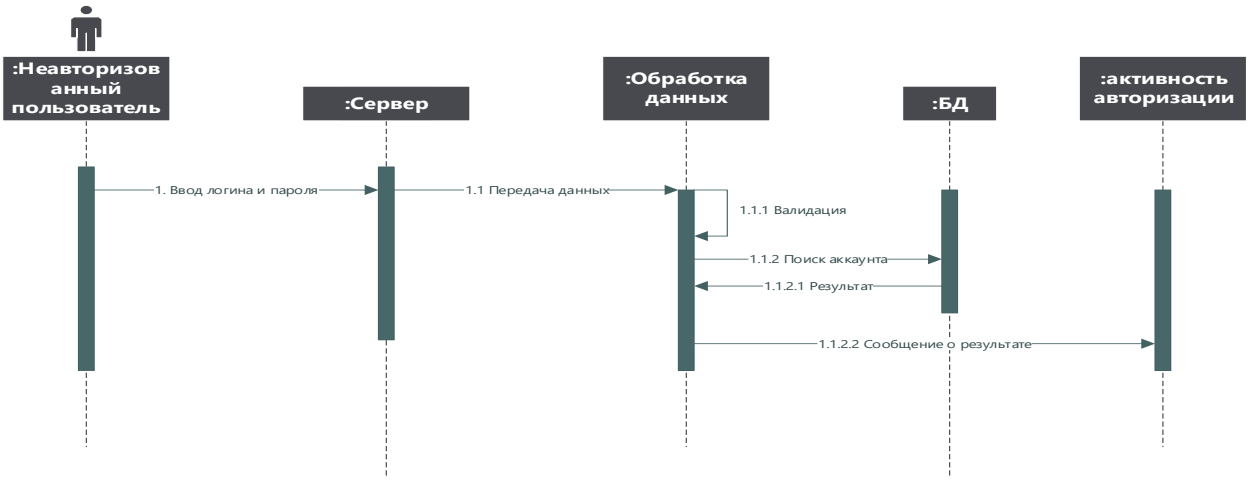

Рисунок 7 – Диаграмма последовательности для прецедента авторизация

Прецедент авторизации отображен на диаграмме последовательности (рисунок 7).

Для разрабатываемого продукта одним из требований является повышенная безопасность. Такое заключение было получено из анализа существующих продуктов в главе 1, где, по итогам анализа, большинство аналогов не обеспечивают надежной защиты мастер пароля.

Исходя из вышесказанного, было принято решение спроектировать модуль авторизации в соответствии с технологией JWT стандарта RFC 7519. JSON Web Token (JWT) — это *JSON* объект, который определен в открытом стандарте RFC 7519. Он считается одним из безопасных способов передачи информации между двумя участниками. Для его создания необходимо определить заголовок (header) с общей информацией по токену, полезные данные (payload), такие как id пользователя, его роль и т. д. и подписи (signature). Схема авторизации на основе JWT показана на рисунке 8.

Безопасность и надежность данной технологии подтверждается использованием её в таких системах как «Сбербанк Онлайн», «ВТБ Онлайн», «Яндекс Деньги» и «Яндекс Касса».

Так же различными реализациями данной технологии пользуются такие сервисы как Google, Facebook. Они применяют её при предоставлении сторонним разработчикам своего API. К примеру, авторизация с помощью аккаунта Google на стороннем сайте работает на основе JSON Web Token.

Рисунок 8 – Схема авторизации на основе JWT токенов

Однако, *ни одна из вышеперечисленных реализации данной технологии не находится в открытом доступе*, видимо, *из-за требований безопасности*.

Доверие, которое оказывается JWT крупными предприятиями, банками и различными корпорациями, позволяет говорить о высокой надежности решений, разработанных на базе этой технологии.

Функциональная диаграмма IDEF0 модуля авторизации показана на рисунке 9. Декомпозиция представлена на рисунке 10.

На диаграмме видно, что входными данными являются логин и пароль пользователя приложения. В результате работы модуля авторизации получается JSON Web Token. Регламентирующими документами, в свою очередь, являются: стандарт RFC 7519 (описывает процедуру генерации JWT), протокол Http (контролирует процесс передачи данных от клиента к серверу и обратно)

и СУБД (которая обеспечивает проверку данных, предоставленных пользователем).

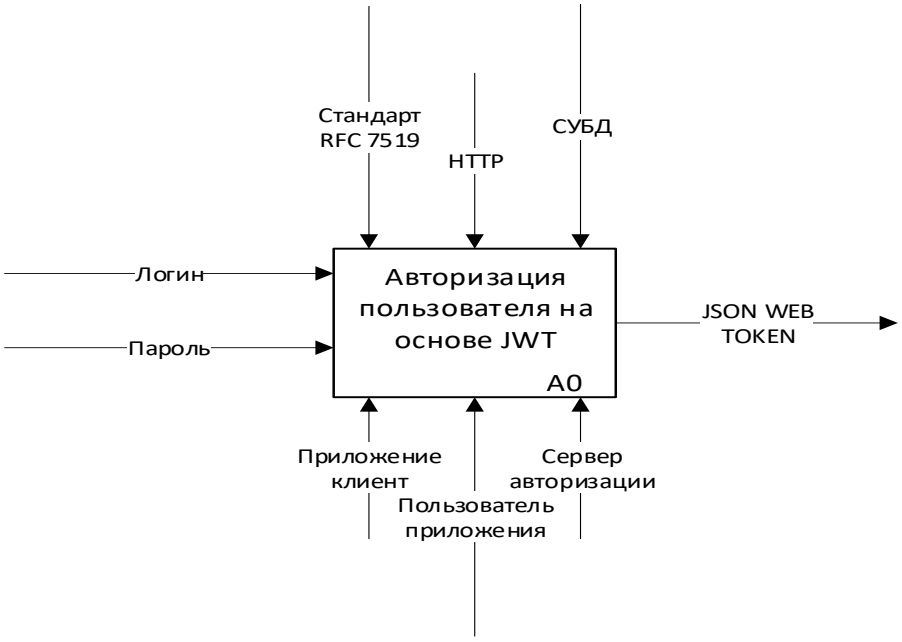

Рисунок 9 – Диаграмма IDEF0 модуля авторизации на основе JWT

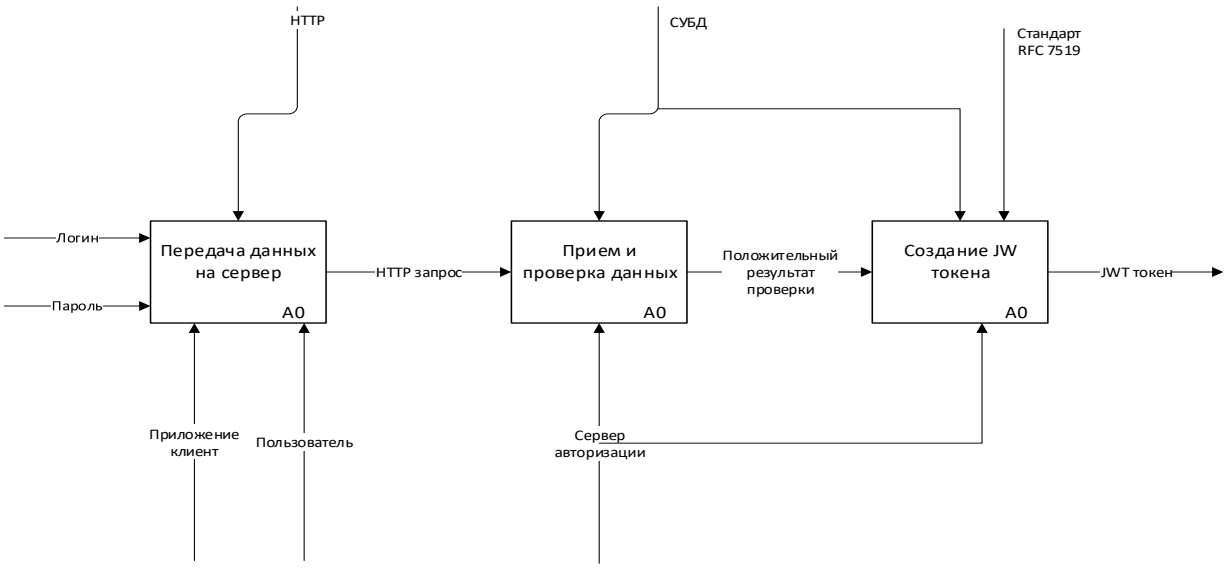

Рисунок 10 – Декомпозиция верхнего уровня модуля авторизации

2.3.2 Модуль логики сервера

Модуль сервера занимается обработкой входящих от клиентов запросов. Так же сервер реализует взаимодействие с базой данной выполняя необходимые операции. Схема обработки входящих HTTP запросов, сопоставление запрашиваемого и существующего контроллера приведена на рисунке 12.

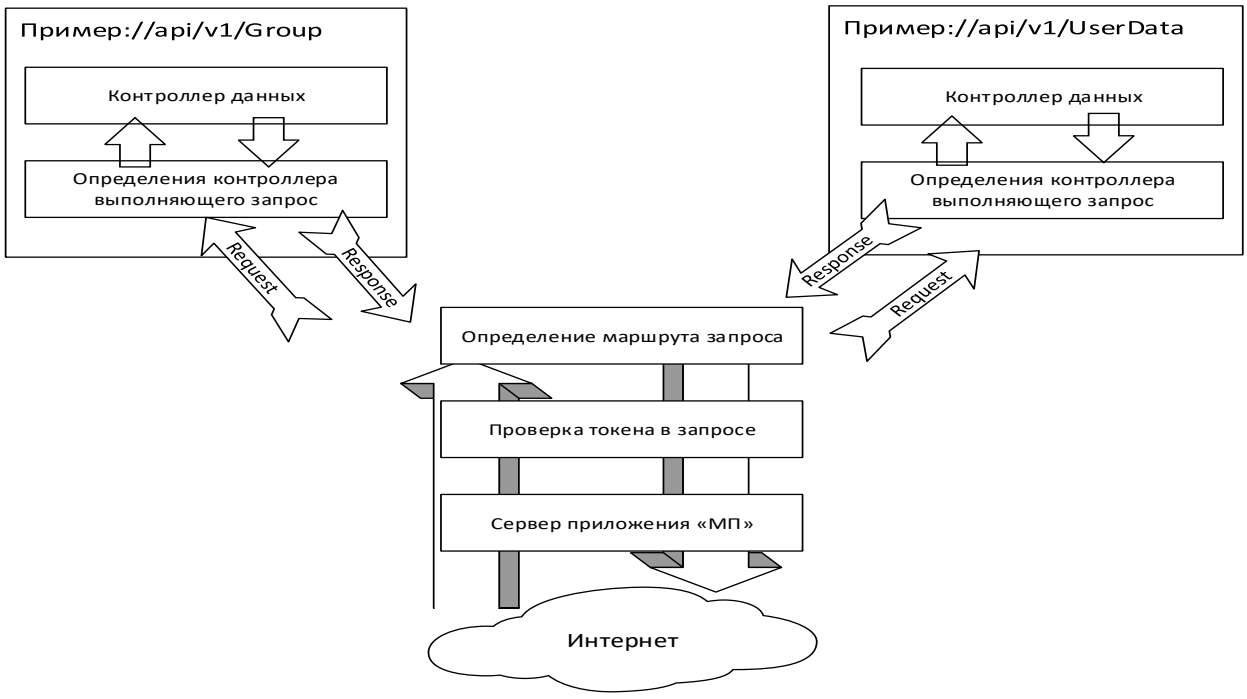

Рисунок 11 – Схема обработки входящих HTTP запросов сервером приложений

2.3.3 Модуль интерфейса пользователя

Данный модуль функционирует на стороне приложения-клиента. Основной задачей данного модуля является предоставление графического интерфейса, реагирующего на действия пользователя и уведомляющего его о возможных проблемах и сбоях в работе приложения.

Модуль интерфейса пользователя не генерирует никаких данных, данные для вывода он получает из модулей нижнего уровня. При получении данных от пользователя, модуль выполняет их первичную проверку с последующей делегацией данных нижним (сетевым и транспортным уровням) приложения. Функциональная модель модуля в нотации IDEF0 представлена на рисунке 12. Из рисунка 12 видно, что модуль интерфейса получает данные для отображения и на выход предоставляет пользователю их графическое представление в интерфейсе. Регламентирующими документами являются правила валидации (проверки) данных (обеспечивают соответствие типов данных) и Material Design (регламентирует соответствие внешнего вида элементов интерфейса и их формы стандарту Material Design). Программная логика

отображения интерфейса и элементы интерфейса преобразуют входные данные в выходные.

Рисунок 12 – Функциональная модель модуля интерфейса в нотации IDEF0

2.3.4 Модуль клиент-серверного взаимодействия

Основными целями данного модуля является обеспечение основного функционала приложения, посредством передачи данных между клиентом и сервером.

Данный модуль построен таким образом, чтобы никак не взаимодействовать с конкретными элементами пользовательского интерфейса. На этапе разработки (кодирования) это может быть достигнуто посредством «инверсии зависимостей». Интерфейс инициирует работу модуля и в результате получает данные или ответ от сервера и отображает их пользователю.

Этапы работы модуля клиент-серверного взаимодействия:

- инициализация модуля, его первичная настройка при запуске приложения;
- инициация работы функций модуля интерфейсом;
- предварительная проверка данных перед отправкой на сервер;
- возвращение результата работы;

Диаграмма работы данного модуля представлена на рисунке 13.

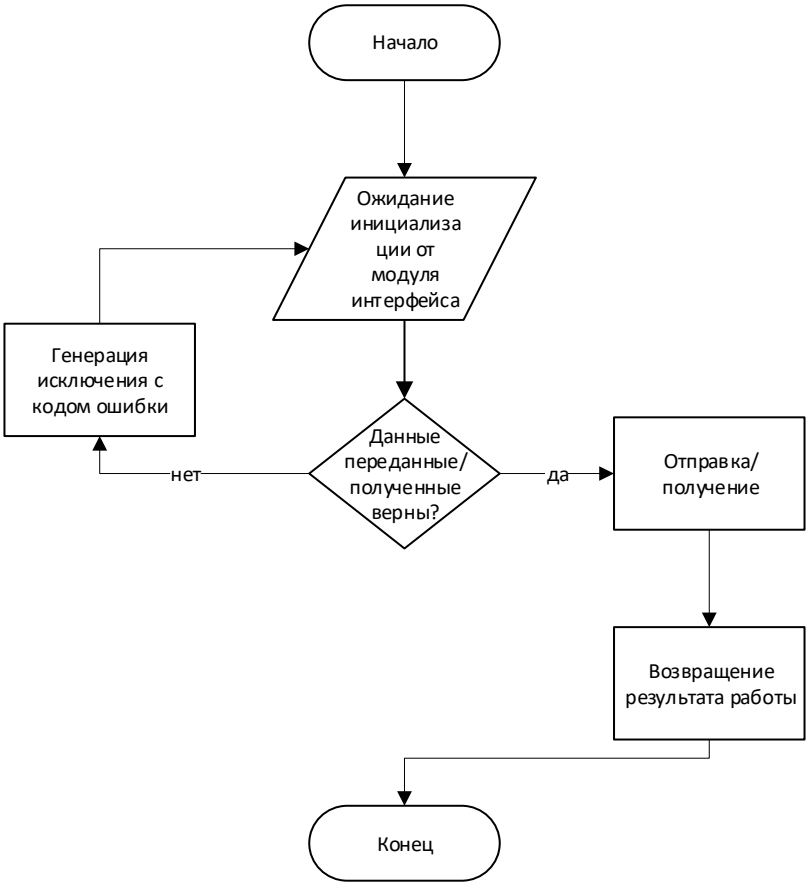

Рисунок 13 – Диаграмма работы модуля клиент серверного взаимодействия

2.3.5 Модуль обработки данных сессии

Основной задачей модуля является хранение JWT токена и предоставление его сетевым модулям приложения. Так же модуль отслеживает время жизни токена и выполняет действия по повторной авторизации или автоматическому выходу пользователя из системы.

2.4 Проектирование базы данных

Исходя из требований модуля авторизации на основе JWT токенов, пользователь должен предоставить аутентификационную пару – логин/пароль. Следовательно, необходима таблица Users.

Так же для обеспечения безопасности пользователя, в таблицу необходимо включить атрибуты, отвечающие за использование двухфакторной аутентификации, блокировку аккаунта и подтверждение владения электронной почтой/телефоном. В дальнейшем, при расширении функционала модуля,

возможно дифференцирование пользователей по ролям (пользователь/организация/администратор). Для этого будет необходима таблица Roles. Одному пользователю соответствует одна роль для чего введем таблицу UserRoles.

Таблица 4 – Атрибуты таблицы AspNetUsers

Атрибут	Описание	Тип данных	NULL
1	2	3	
Id (PK)	Уникальный идентификатор	Nvarchar(450)	нет
UserName	Имя пользователя	Nvarchar(256)	да
NormalizedUserName	Нормализованное относительно регистра имя пользователя	Nvarchar(256)	да
Email	Электронный адрес	Nvarchar(256)	да
NormalizedEmail	Нормализованное относительно регистра Email	Nvarchar(256)	да
EmailConfirmed	Подтвержден ли адрес электронной почты	bit	нет
PasswordHash	Хеш пароля пользователя	Nvarchar(max)	да
PhoneNumber	Номер телефона	Nvarchar(max)	да
PhoneNumberConfirmed	Подтвержден ли номер телефона	bit	нет
TwoFactorEnabled	Включена ли двухфакторная аутентификация	bit	нет
AccessFailedCount	Количество неуспешных попыток входа	int	нет
LockoutEnabled	Включение блокировки аккаунта	bit	нет
LockoutEnd	Дата/время окончания блокировки	datetimeoffset(7)	да

Таблица 5 – Атрибуты таблицы AspNetRoles

Атрибут	Описание	Тип данных	NULL
Id (PK)	Уникальный идентификатор	Nvarchar(450)	нет
Name	Имя пользователя	Nvarchar(256)	да
NormalizedName	Нормализованное относительно регистра имя пользователя	Nvarchar(256)	да

Таблица 6 – Атрибуты таблицы AspNetUserRoles

Атрибут	Описание	Тип данных	NULL
UserId (Clustered PK)	Id пользователя	Nvarchar(450)	нет
RoleId (Clustered PK)	Id роли	Nvarchar(450)	нет

Сущности, передаваемые посредством модуля клиент-серверного взаимодействия представлены в таблицах 7,8

Таблица 7 – Атрибуты таблицы Group

Атрибут	Описание	Тип данных	NULL
Id	Id группы	Nvarchar(450)	нет
Name	Название группы	Nvarchar(256)	нет
Preview	Иконка группы	Byte	да
UserId	Идентификатор пользователя-владельца группы	Nvarchar(256)	нет

Таблица 8 – Атрибуты таблицы UserData

Атрибут	Описание	Тип данных	NULL
Id	Id аутентификационной пары	Nvarchar(450)	нет
Login	Логин	Nvarchar(256)	нет
Password	Пароль	Nvarchar(256)	да
Source	Сайт учетной записи	Nvarchar(450)	нет
SharedBy	Поделился с	Nvarchar(max)	

Также для обеспечения функции обновления и проверки совместимости требуется сущность ServerInfo.

Таблица 9 – Атрибуты таблица Info

Атрибут	Описание	Тип данных	NULL
Id	Id аутентификационной пары	Nvarchar(450)	нет
ActualClientViersionMin	Логин	Nvarchar(20)	нет
ActualClientVersionMax	Пароль	Nvarchar(20)	нет

Итоговая диаграмма IDEF1X базы данных для модуля авторизации приложения «Менеджер паролей» отображена на рисунке в Приложении А.

2.5 Требования к программному продукту

2.5.1 Общие требования

а) Продукт должен состоять из программных модулей, во-первых, это позволит обеспечить легкое обновление программного обеспечения (по средством замены устаревшего программного модуля на более новый), во-вторых, такая

структура информационной системы будет максимально полно совпадать с функциональными модулями, описанными в пункте 2.3.

б) Продукт должен быть надежен, соединение клиентов с сервером должно быть защищено.

в) Продукт должен поддерживать различные СУБД и не иметь привязки к конкретной. В первую очередь данное требование вытекает из требования кроссплатформенности. Оптимальным решением при размещении на серверах с операционной системой Windows Server может стать СУБД MS SQL Server. Однако данная СУБД не доступна для операционных систем старше Ubuntu 18 и имеет достаточно высокие минимальные требования к аппаратной части сервера, как следствие более разумным выбором может являться другая СУБД.

г) База данных, используемая программным продуктом, должна использовать актуальные и устойчивые ко взлому методы шифрования.

д) Интерфейс должен быть простым и понятным с низкой сложностью освоения и поощрять знания и умения людей, которые работали со схожим программным продуктом.

е) Цветовая гамма, используемая в приложении, должна быть приятной и не раздражающей.

2.5.2 Требования к лингвистическому обеспечению

Основным языком приложения является русский язык. Интерфейс должен включать элементы, сообщения и подсказки только на русском языке.

2.5.3 Требования к информационному обеспечению

В связи с необходимость хранить большой объем информации, поступающий от множества клиентов, на стороне сервера необходимо использовать одну из современных и надежных реляционных СУБД, таких как MS SQL Server, MySQL, PostgreSQL.

2.5.4 Требования к техническому обеспечению

Основная нагрузка по обработке, поиску и хранению лежит на серверной

части приложения «Менеджер паролей». В связи с этим программный продукт не требует значительных вычислительных мощностей и как следствие поддерживается на любом устройстве. Однако, в целях обеспечения дополнительной безопасности из перечня поддерживаемых систем исключены системы снятые с поддержки производителя: Windows 7 и старше, т. к. для этих систем не выпускаются обновления и заплатки уязвимостей.

Таким образом минимальные требования к устройству пользователя следующие:

- ОС: Windows 10, Ubuntu 16 LTS и новее, Android 7 и новее;
- Процессор: Intel Core Duo, Amd Phenom, Arm7a и новее;
- Объем оперативной памяти 128 Мб;
- Место на накопителе 100 Мб;
- Доступ в интернет.

3 ОПИСАНИЕ ПРИЛОЖЕНИЯ «МЕНЕДЖЕР ПАРОЛЕЙ»

3.1 Выбор методологии разработки программного обеспечения

В начале этапа проектирования необходимо определить методологию разработки программного обеспечения. Исходя из требований к продукту, определенных на предыдущем этапе, и объеме планируемых работ, оптимальными являются методологии, которые смогут обеспечить качество и скорость разработки программного продукта, а также гибкость к изменениям требований.

Одной из таких методологий является «Экстремальное программирование» (XP). Она включает в себя следующие особенности, которые позволят полноценно реализовать требования к продукту:

а) Короткий цикл обратной связи включает в себя разработку через тестирование и парное программирование, эти компоненты способствуют увеличению качества выходного продукта.

б) Непрерывный процесс включает в себя непрерывную интеграцию, рефакторинг и небольшие релизы. Непрерывная интеграция и небольшие релизы позволят увидеть реальную полезность и подтвердить необходимость разработки продукта еще на ранних этапах. А рефакторинг кода обеспечит качественную кодовую базу, которую будет легко поддерживать и модифицировать.

Рисунок 14 – Практики «Экстремального программирования»

3.2 Выбор средств разработки

Исходя из требований к программному продукту и перечня функций, требуемых к реализации, средства разработки должны предоставлять возможность разработки как клиентской, так и серверной логики, иметь хорошую интеграцию с существующими реляционными базами данных, а также поддержку кроссплатформенных приложений. Так же, в соответствии с выбранной методологией разработки, инструментарий должен в полной мере обеспечивать поддержку разработки через тестирование.

Подходящими кандидатами являются следующие языки программирования:

- Python;
- Java;
- C#.

Для выбора наилучшего варианта сгруппируем особенности данных языков, связанные с решаемой задачи в таблицу. Важными особенностями в разработке модуля авторизации являются его надежность и универсальность, удобство запросов к базе данных, кроссплатформенность, быстродействие.

Требование кроссплатформенности исходит из необходимости исключить зависимость компонента серверной логики и базы данных от конкретной серверной среды (Windows, Unix), т. к. она будет неудобной и дорогостоящей. Сравнение средств разработки приведено в таблице 10.

Среда .NET Core 2.2/3.0 являются полностью кроссплатформенной, а использование стандарта .NET Standart 2.0 позволяет создавать кроссплатформенные библиотеки совместимые с различными средами .NET.

Языком написания является C#, для работы с БД используется EF Core 6 (версия EF6 для платформы .NET Core). Это позволяет взаимодействовать с таблицами базы данных как с обычными моделями (классами). Так же для расширения функционала можно использовать LINQ (Language Integrated Query) – язык интегрированных запросов, позволяющий одинаково взаимодействовать с любыми перечисляемыми объектами (LINQ to List, Array, Entities).

Так же EF Core 6 поддерживает любые СУБД без различий в специфике. Т. е. код приложения, написанный с помощью EF и LINQ, не зависит от конкретной СУБД, что позволяет менять конкретную СУБД по требованию заказчика.

Visual Studio 2019 является мощным инструментом, призванным помочь разработчикам, и несмотря на наличие коммерческой версии, в командах менее 5 человек, можно использовать бесплатную Community версию, не обладающую ограничениями.

Таблица 10 – Сравнение инструментов разработки

Наименование	Вариант 1	Вариант 2	Вариант 3
Язык программирования	C#	Python	Java
Кроссплатформенность	да	Не полная	да
Поддержка Unit тестов	да	да	да
Взаимодействие с БД	Entity Framework Core 6	SQL Alchemy	SQL
Клиент-серверное взаимодействие	да	да	да
Среда	.net core 2.2/3.0	Python 3.7	Java 11
СУБД	Любая	MySQL	MySQL, Postgres, Oracle DB
IDE	VS 2019, VS Code	PySharm, VS Code	IntelliJ IDEA

Java является известным и мощным языком программирования, однако отсутствие в нем ORM технологий, подобных Entity Framework, существенно повлияет на качество и скорость разработки. Запросы написанные на SQL будут привязаны к специфике конкретной СУБД, и её последующая замена будет невозможна или затруднительна, что противоречит требованию расширяемости. Так же IDE для полноценной работы с Java является дорогим удовольствием, стоимость IntelliJ IDEA 3350 руб./месяц или же 33 500 руб. в год за одну лицензию.

Python современный кроссплатформенный язык. В его состав по умолчанию не входит какая-либо ORM система для работы с БД, однако существуют сторонние расширения такие как SQL Alchemy. К недостаткам таких решений можно отнести их узкую направленность (SQL Alchemy работает

только с MySQL) и открытую кодовую базу, которая может кардинально изменяться с течением времени, что может затруднить поддержку существующего кода. Так же, полная кроссплатформенность не является частью языка по умолчанию, и, к примеру, разработка клиентской логики под операционную систему Android может стать довольно затруднительной.

Таким образом, наиболее подходящим решением является разработка программы на базе среды .NET Core с использованием ЯП C#.

3.3 Разработка модулей

Для реализации требования модульности был выбран архитектурный паттерн MVC (приложение-сервер) и комбинации архитектур DDD и Onion (приложение-клиент).

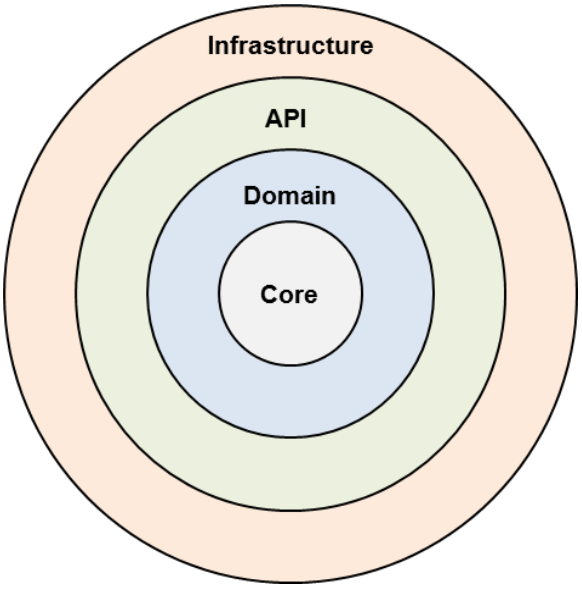

Рисунок 15 – Архитектура приложения построенная на основе Domain Driven Design и Onion паттернов

Каждый слой приложения клиента согласно DDD представляет собой отдельную сборку (.dll). Согласно паттерну Onion зависимости между слоями должны распространяться только сверху в низ, также допускаются внутри слоевые взаимодействия.

Итоговая схема взаимодействия модулей приложения-клиента представлена на рисунке 16.

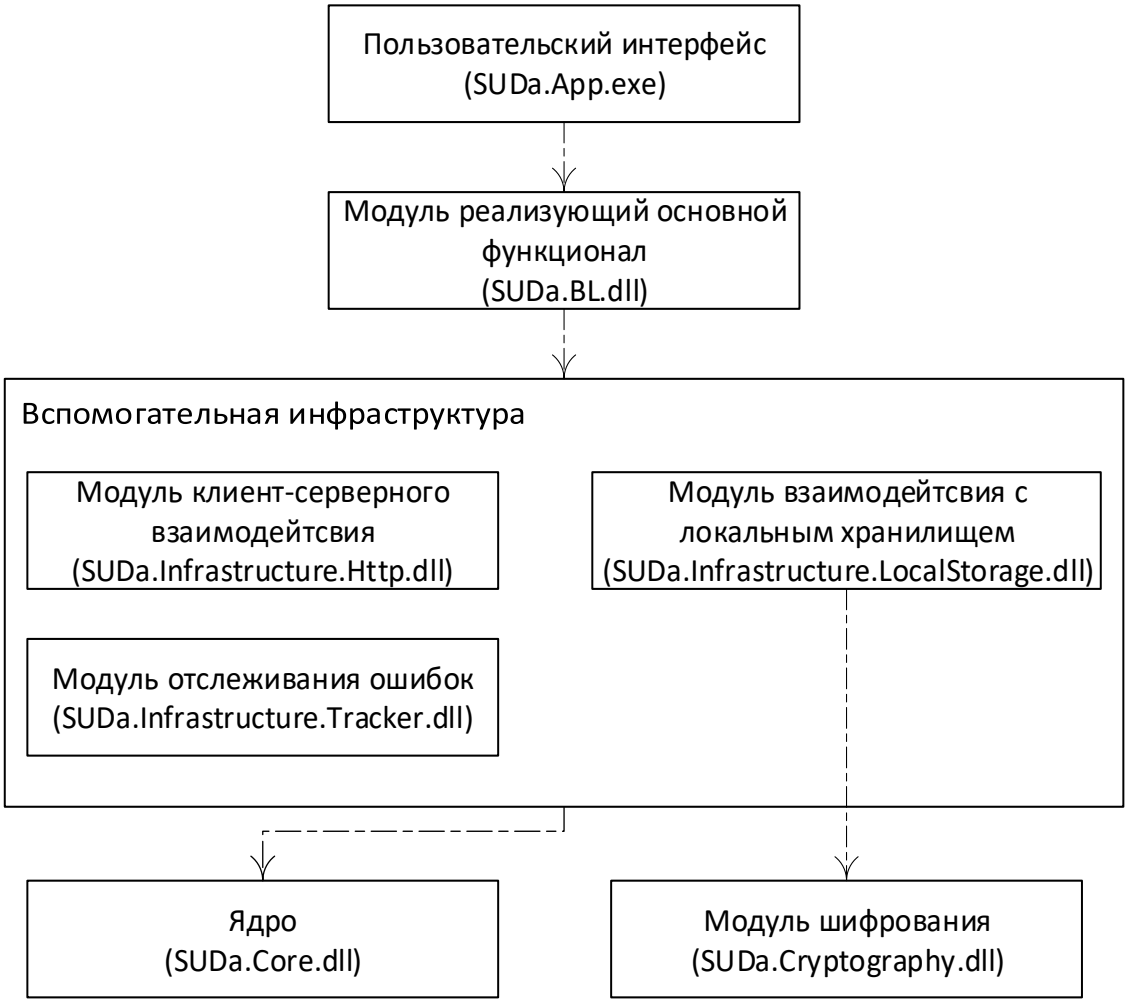

Рисунок 16 –Взаимодействие модулей приложения-клиента

3.4 Пример реальной эксплуатации

Для уменьшения стоимости эксплуатации было принято решение расположить сервер приложения, СУБД, Web-приложение на одном виртуальном сервере, предоставляемом хостингом. Проблема заключается в том, что сервер приложения, построенный на базе ASP NET Core, использует для обработки http(s) запросов web сервер Kestrel, который перехватывает все входящие http запросы (т. е. не позволяет другим http сервисам принимать предназначенные им пакеты). Схема обработки http запросов, в результате простого размещения всех компонентов на web сервере показана на рисунке 17.

Решением данной проблемы является использование обратного прокси сервера, задачей которого будет являться определение и перенаправление входящих запросов адресату.

Используя один IP адрес и присылая запросы на разные порты, можно обращаться к интересующемуweb-сервису (рисунок 18).

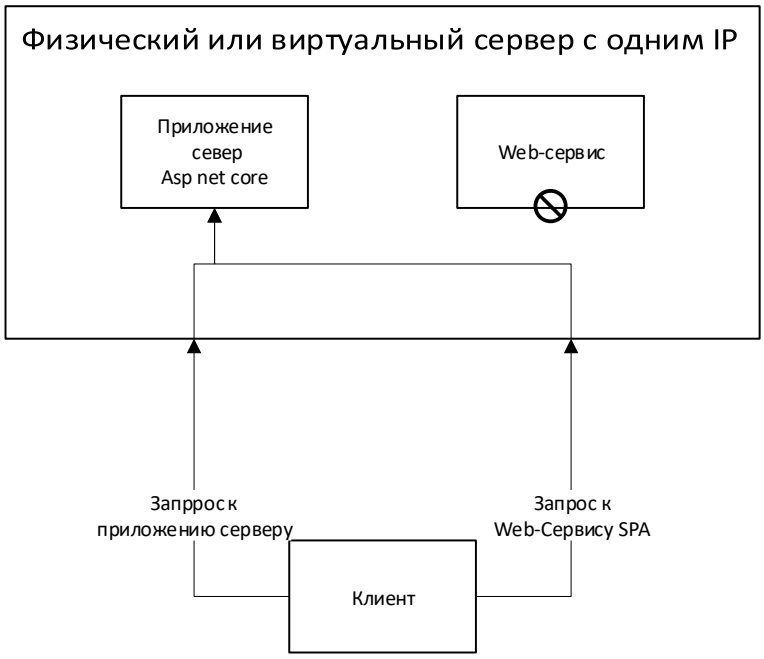

Рисунок 17 – Схема обработки http забросов без прокси сервера

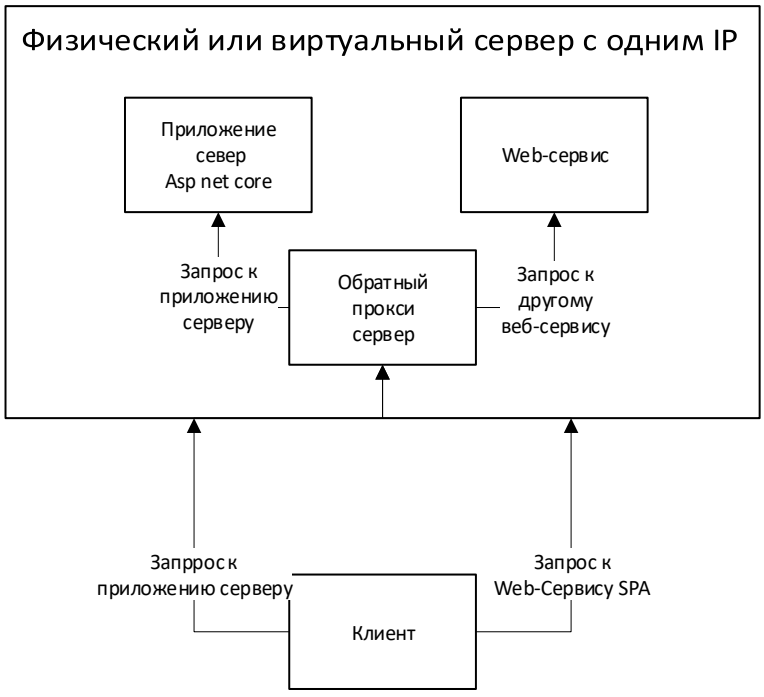

Рисунок 18 – Схема обработки http забросов с использованием обратного прокси сервера

3.5 Описание экранных форм

Интерфейс реализован с использованием современного и интуитивно понятного Material Design.

В нижнем правом углу в выпадающем меню можно создать новый аккаунт, удалить существующий.

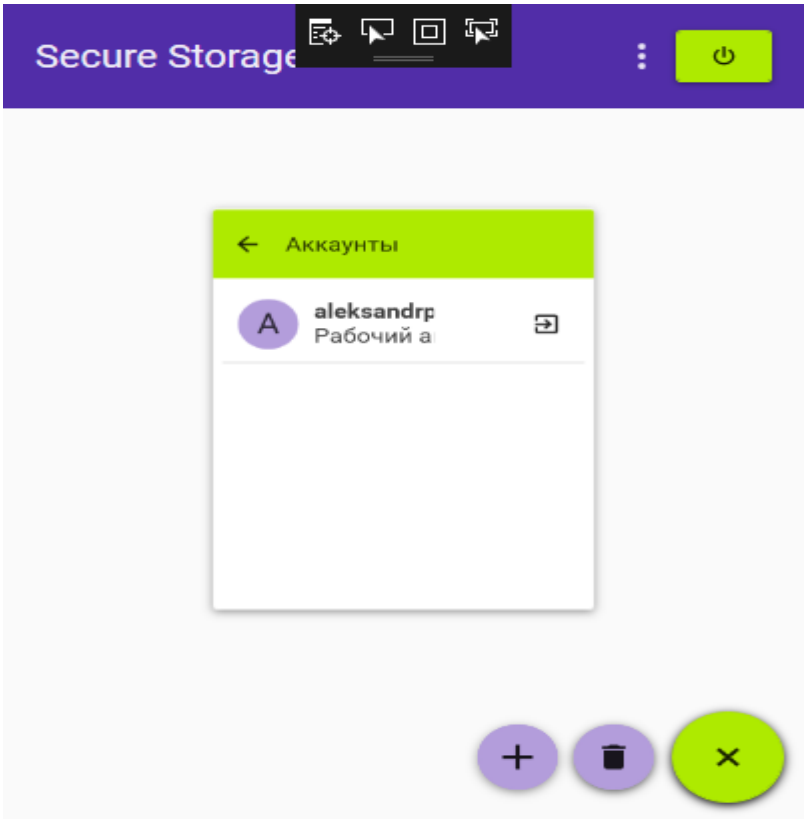

Рисунок 19 – Меню действий над аккаунтами

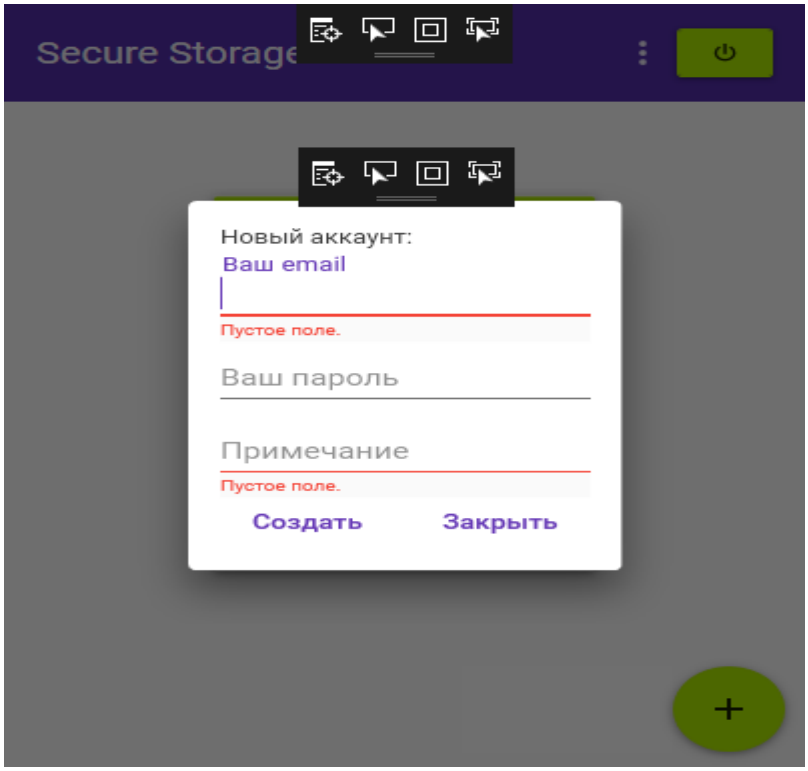

Рисунок 20 – Создание нового аккаунта

В случае возникновения ошибки в результате выполнения запроса, пользователь получает уведомление. Пример уведомления показан на рисунке 21.

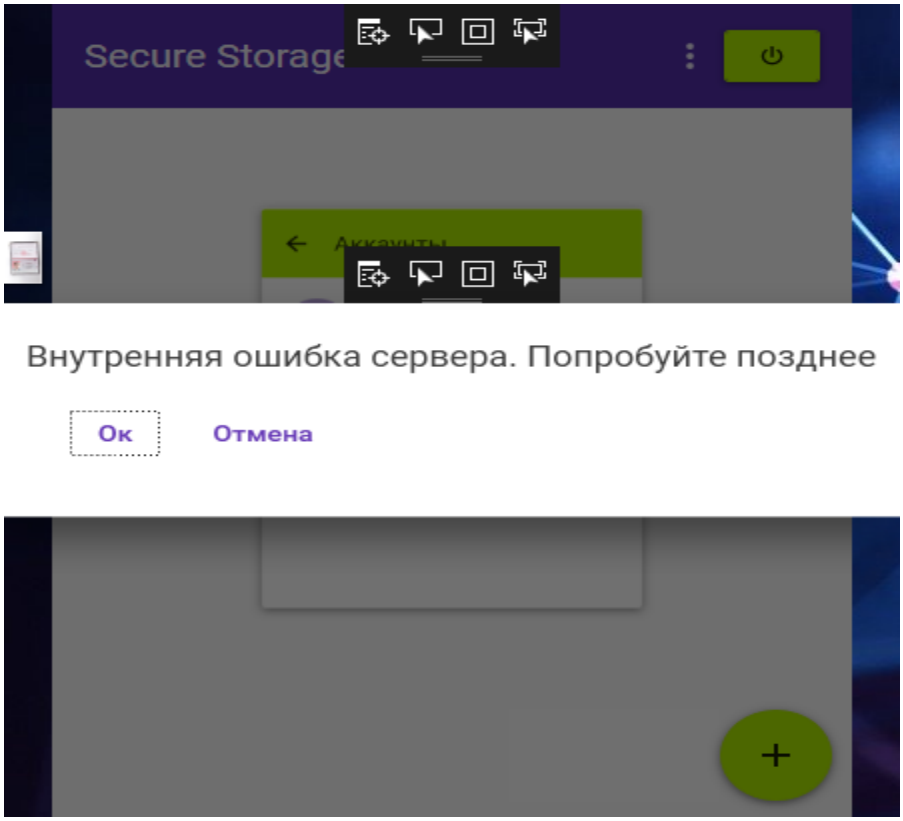

Рисунок 21 – Уведомление пользователя о возникших ошибках

4 ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ КРОССПЛАТФОРМЕННОГО КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ «МЕНЕДЖЕР ПАРОЛЕЙ»

Клиент-серверные приложения становятся всё более популярными во всех сферах деятельности человека и, в свою очередь, хранят и обрабатывают много приватных данных, что неизбежно приводит к ряду проблем в области безопасности. Для решения данных проблем был сформирован международный проект по обеспечению безопасности WEB-приложений (OWASP).

4.1 Модель угроз информационной безопасности

Модель угроз информационной безопасности – это описание существующих угроз ИБ, их актуальности, возможности реализации и последствий.

Для построения модели угроз информационной безопасности информационной системы используются методики и каталоги угроз из официального стандарта ГОСТ Р 51275-2006, методических документов ФСТЭК. Сам процесс построения модели представляет собой последовательность следующих операций:

- выявление источников угроз информационной безопасности;
- определение критически важных активов;
- определение актуальных угроз безопасности информационной системы и способов их реализации;
- оценка материального ущерба и других последствий возможной реализации угроз;

Согласно «Методике определения угроз безопасности информации в информационных системах» (ФСТЭК), модель угроз безопасности информации должна содержать следующие основные разделы:

- описание информационной системы и особенностей ее функционирования;
- возможности нарушителей (модель нарушителя);
- актуальные угрозы безопасности информации;

Для приложения, состоящего из двух постоянно взаимодействующих элементов – клиента и сервера, необходимо рассмотреть модели угроз для клиента так и для и сервера.

Приложение обладает следующими характеристиками:

- соответствие информационной системы заданным стандартам;
- использование защищенных протоколов обмена данными;
- использование средств антивирусной защиты;
- межсетевое экранирование;
- выполнение процедур резервного копирования и другие.

Для каждого из активов были определены актуальные угрозы безопасности информационной системы, которые в совокупности с источниками угроз представлены на *диаграмме вариантов использования* (рисунок 22).

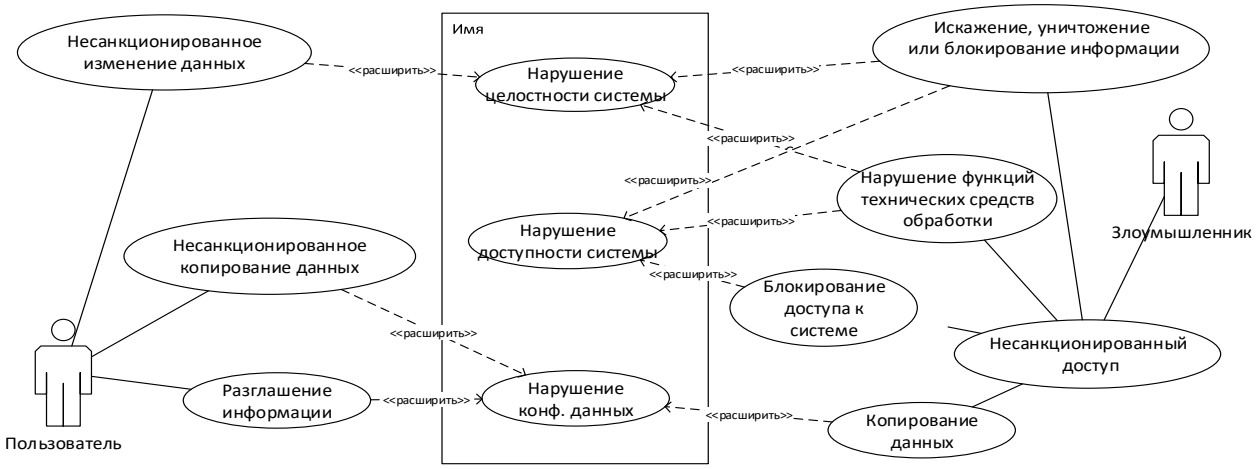

Рисунок 22 – Диаграмма вариантов использования. Модель угроз

Варианты реализации угроз приведены ниже.

Рисунок 23 – Способы реализации угрозы «Разглашение информации»

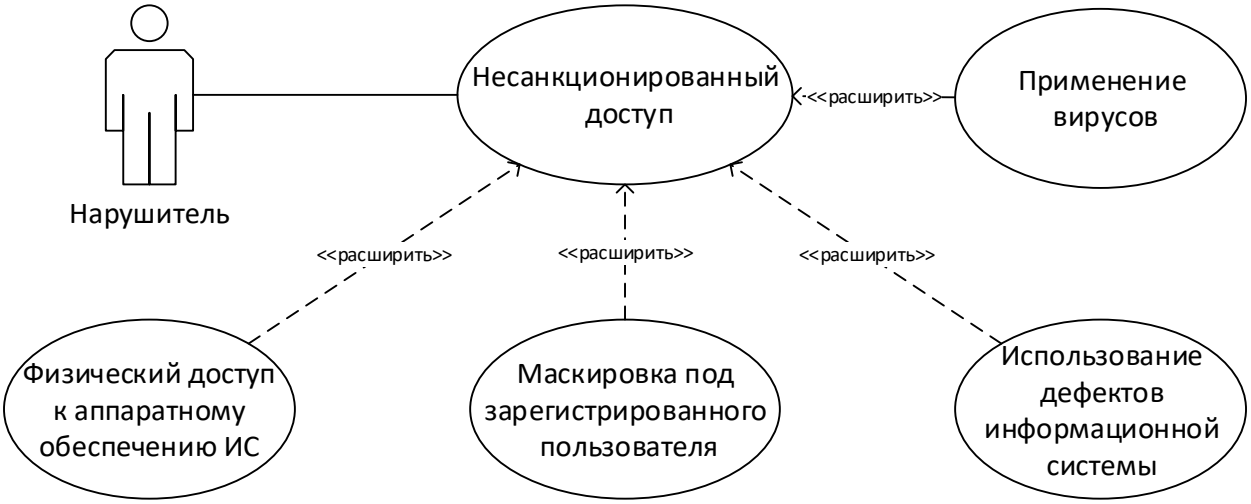

Рисунок 24 – Способы реализации угрозы «Несанкционированный доступ»

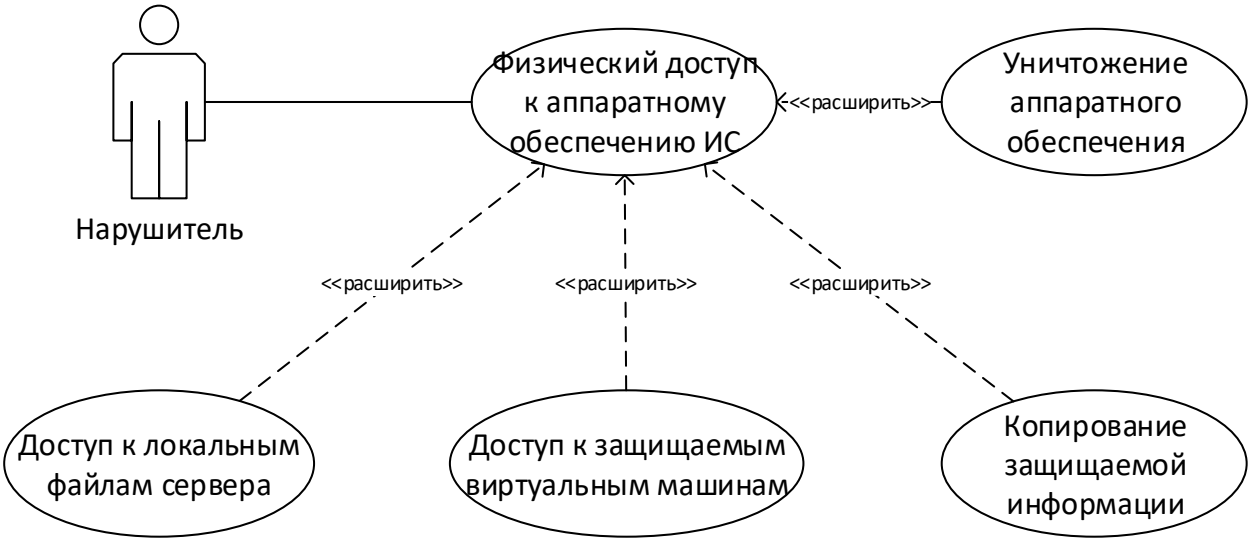

Рисунок 25 – Варианты реализации угрозы «Физический доступ»

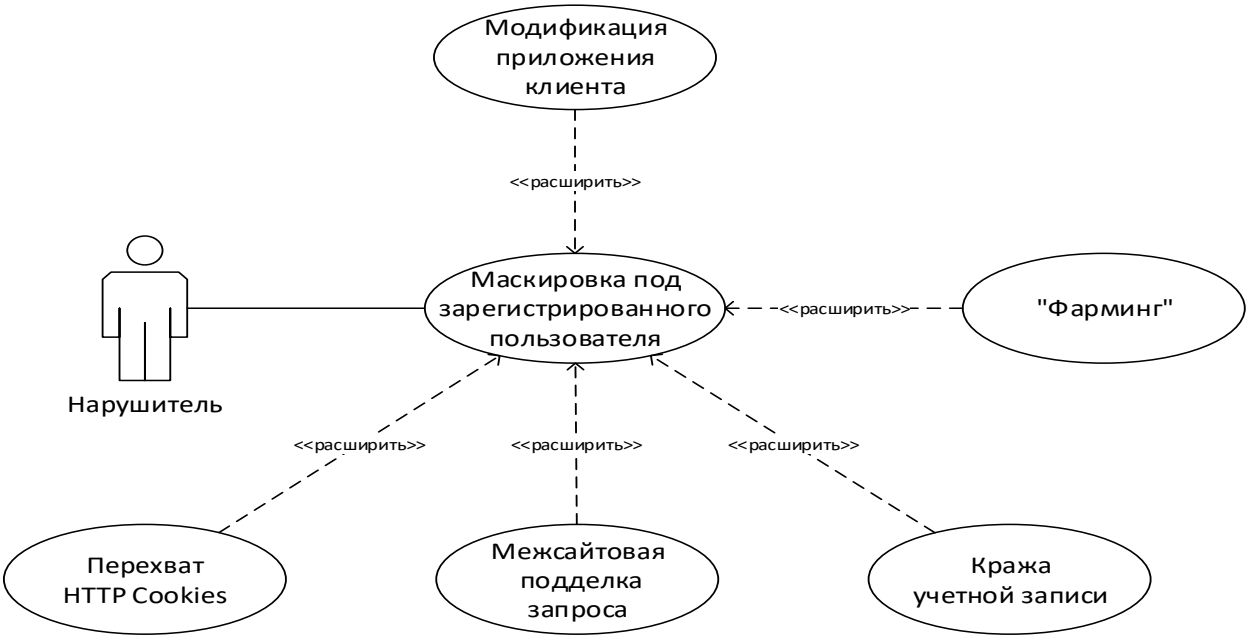

Рисунок 26 – Варианты реализации угрозы «Маскировка под пользователя»

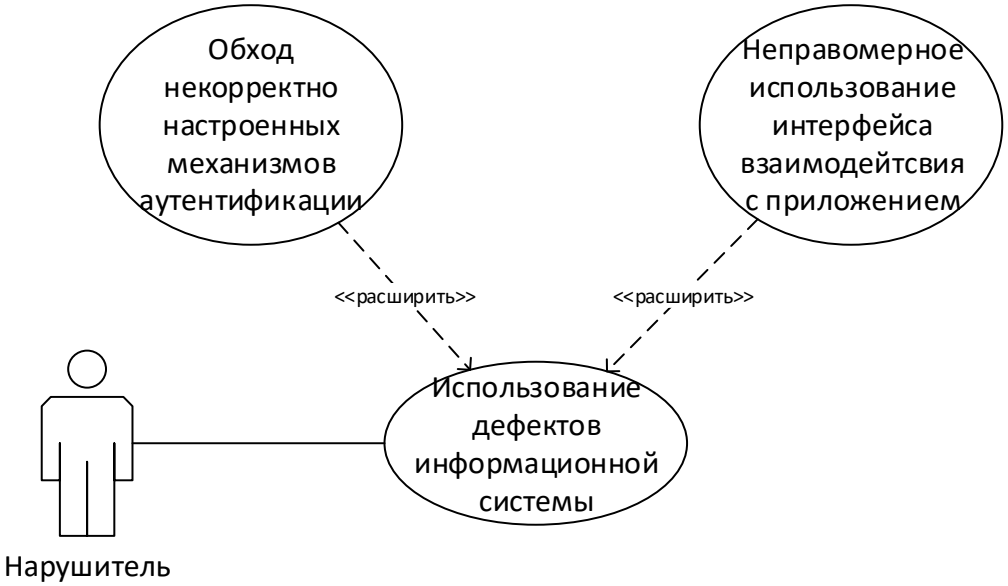

Рисунок 27 – Варианты реализации угрозы «Использование дефектов ИС»

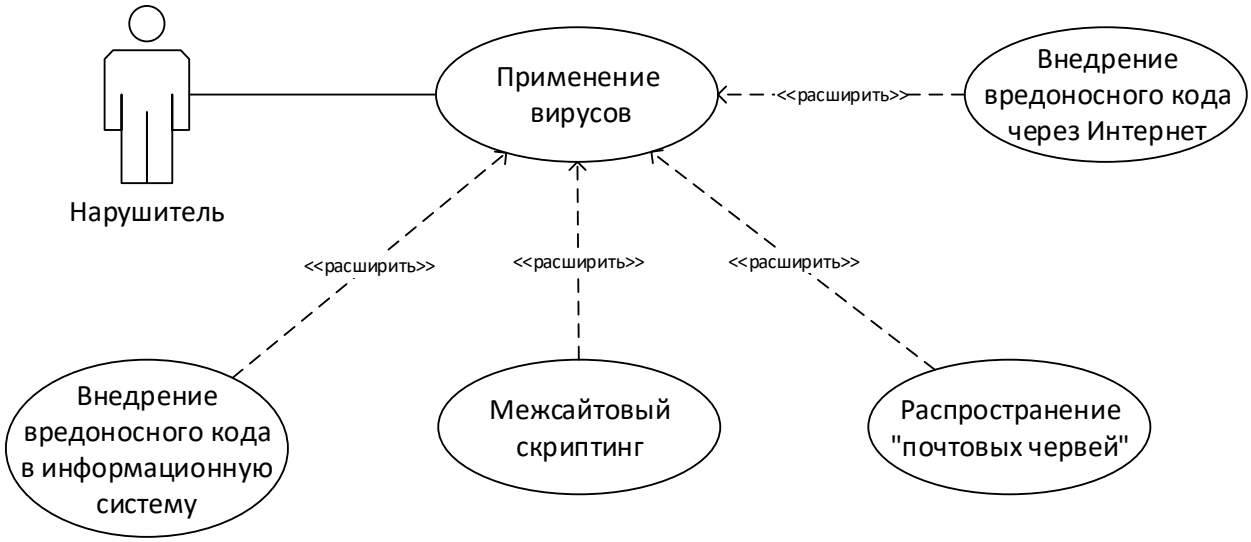

Рисунок 28 – Варианты реализации угрозы «Применение вирусов»

Таблица 11 – Варианты реализации угроз

Название угрозы	Реализация угрозы	Объект воздействия	Нарушение конфиденциальности (ущерб)	Нарушение целостности (ущерб)	Нарушение доступности (ущерб)
1	2	3	4	5	6
Внедрение вредоносного кода в ИС	внедрение нарушителем в дискредитируемую информационную систему вредоносного кода	Системное программное обеспечение, прикладное программное обеспечение	высокий	высокий	низкий

Продолжение таблицы 11

1	2	3	4	5	6
Восстановления аутентификационной информации	подбор (например, путём полного перебора или перебора по словарю) аутентификационной информации дискредитируемой учётной записи пользователя в системе.	Системное программное обеспечение, микропрограммное обеспечение, учётные данные пользователя	средний	—	низкий
Доступ к локальным файлам сервера при помощи URL	передачи нарушителем дискредитируемому браузеру запроса на доступ к файловой системе пользователя вместо URL запроса.	Сетевое программное обеспечение	средний	—	—
Доступ/перехват/изменения HTTP cookies	осуществление нарушителем несанкционированного доступа к защищаемой информации (учётным записям пользователей, сертификатам и т. п.), содержащейся в cookies-файлах	Прикладное программное обеспечение, сетевое программное обеспечение	средний	—	средний
Межсайтовый скриптинг	внедрение нарушителем участков вредоносного кода на сайт дискредитируемой системы таким образом, что он будет выполнен на рабочей станции просматривающего этот сайт пользователя.	Сетевой узел, сетевое программное обеспечение	средний	средний	–
Неправомерное использования интерфейса взаимодействия	осуществление нарушителем деструктивного программного воздействия на API в целях реализации функций, неправомерные вредоносные действия от имени дискредитированного пользователя.	Системное программное обеспечение, прикладное программное обеспечение,	средний	средний	средний

Продолжение таблицы 11

1	2	3	4	5	6
Несанкционированный доступа к защищаемым виртуальным машинам	осуществление деструктивного программного воздействия на защищаемые виртуальные машины	Виртуальная машина	средний	средний	средний
Несанкционированное копирование защищаемой информации	неправомерное получения нарушителем копии защищаемой информации путём проведения последовательности неправомерных действий, включающих: несанкционированный доступ к защищаемой информации, копирование найденной информации на съёмный носитель.	Объекты файловой системы, машинный носитель информации	средний	—	—
Несанкционированное создания учётной записи пользователя	создание нарушителем в системе дополнительной учётной записи пользователя и её дальнейшего использования в собственных неправомерных целях	Системное программное обеспечение	средний	средний	средний
Обход некорректно настроенных механизмов аутентификации	получение нарушителем привилегий в системе без прохождения процедуры аутентификации за счёт выполнения действий, нарушающих условия корректной работы	Системное программное обеспечение, сетевое программное обеспечение	средний	средний	средний
Перехват вводимой и выводимой на периферийные устройства	осуществления нарушителем несанкционированного доступа к информации, вводимой и выводимой на периферийные устройства	Системное программное обеспечение, прикладное программное обеспечение, аппаратное обеспечение	средний	—	—

Продолжение таблицы 11

1	2	3	4	5	6
Подмена действия пользователя путём обмана	выполнение неправомерных действий в системе от имени другого пользователя с помощью методов социальной инженерии	Прикладное программное обеспечение, сетевое программное обеспечение	средний	средний	средний
Кража учётной записи доступа к сетевым сервисам	неправомерное ознакомления нарушителем с защищаемой информацией пользователя путём получения информации идентификации/аутентификации, соответствующей учётной записи доступа пользователя к сетевым сервисам (социальной сети, облачным сервисам и др.), с которой связан неактивный/несуществующий адрес электронной почты.	Сетевое программное обеспечение	средний	—	средний
Распространение «почтовых червей»	нарушение безопасности защищаемой информации пользователя вредоносными программами, скрытно устанавливаемыми при получении пользователями системы электронных писем, содержащих вредоносную программу типа «почтовый червь», а также невольного участия в дальнейшем противоправном распространении вредоносного кода.	Сетевое программное обеспечение	средний	средний	средний

Продолжение таблицы 11

1	2	3	4	5	6
«Фарминг»	неправомерное ознакомления нарушителем с защищаемой информацией (в т. ч. идентификации/аутентификации) пользователя путём скрытного перенаправления пользователя на поддельный сайт (выглядящий одинаково с оригинальным), на котором от дискредитируемого пользователя требуется ввести защищаемую информацию.	Рабочая станция, сетевое программное обеспечение, сетевой трафик	средний	—	—
«Фишинга»	неправомерное ознакомления нарушителем с защищаемой информацией (в т.ч. идентификации/аутентификации) пользователя путём убеждения его с помощью методов социальной инженерии (в т.ч. посылкой целевых писем (т.н. spear—phishing attack), с помощью звонков с вопросом об открытии вложения письма, имитацией рекламных предложений (fake offers) или различных приложений (fake apps)) зайти на поддельный сайт (выглядящий одинаково с оригинальным), на котором от дискредитируемого пользователя требуется ввести защищаемую информацию	Рабочая станция, сетевое программное обеспечение, сетевой трафик	средний	—	—

Продолжение таблицы 11

1	2	3	4	5	6
Несанкционированной модификации защищаемой информации	нарушение целостности защищаемой информации путём осуществления нарушителем деструктивного физического воздействия на машинный носитель информации или деструктивного программного воздействия	Объекты файловой системы	—	средний	—
Внедрение вредоносного кода за счет посещения зараженных сайтов	осуществление нарушителем внедрения вредоносного кода в компьютер пользователя при посещении зараженных сайтов.	Сетевое программное обеспечение	средний	средний	средний

4.2 Модель нарушителя информационной безопасности

Модель нарушителя информационной безопасности – это набор предположений об одном или нескольких возможных нарушителях информационной безопасности, их квалификации, их технических и материальных средствах и т. д.

Построим модели возможных нарушителей для приложения «Менеджер паролей» исходя из определенных моделей угроз.

Таблица 12 – Категории нарушителей

Нарушители	Категория нарушителя	Потенциал нарушителя	Возможная мотивация	Предполагаемые возможности
1	2	3	4	5
Пользователи	Внутренний нарушитель	низкий	Непредумышленное распространение личной информации	Полный контроль над своими данными в системе, возможность безвозвратного удаления данных.Возможность сообщить третьим лицам информацию касательно хранимых данных, создав тем самым интерес

Продолжение таблицы 12

1	2	3	4	5
Администратор сетевой части приложения	Внутренний	высокий	Неосторожность, Получение личной выгоды от продажи/удаления открытых данных пользователя	Возможность удаления базы данных, что приведет к потере части данных и нарушению доступности сетевой части Возможность чтения открытых данных пользователя
Лица, обеспечивающие функционирование информационных систем или обслуживающих инфраструктуру оператора	Внутренний нарушитель	низкий	Причинение имущественного ущерба путем обмана или злоупотребления доверием. Непреднамеренные, неосторожные или неквалифицированные действия.	Возможность получить информацию об уязвимостях отдельных компонент информационной системы, опубликованную в общедоступных источниках Возможность получить информацию о методах и средствах реализации угроз безопасности информации (компьютерных атак), опубликованных в общедоступных источниках, и (или) самостоятельно осуществлять создание методов и средств реализации атак и реализацию атак на информационную систему
Третьи лица, нарушители	Внешний нарушитель	Средний	Получение личной выгоды Любопытство	Воздействие на пользователя ИС с целью получения данных Программные/вредоносные воздействия на ИС

5 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

5.1 Безопасность

5.1.1 Анализ эргономики программного обеспечения

5.1.1.1 Общие сведения о понятии эргономики

В настоящее время, сохраняется тенденция к увеличению сложности программного обеспечения, что, в свою очередь, увеличивает требования к такому параметру, как удобство использования, или эргономичность.

Эргономичность – эффективность системы в эргономике. Под эффективностью понимается наибольшая производительность при наименьшей вероятности ошибки.

5.1.2 Анализ эргономики программного продукта «Менеджер паролей»

«Менеджер паролей» - программный продукт разрабатываемы в рамках выпускной квалификационной работы. Назначение данного продукта является оптимизация процессов управления аутентификационными парами логин/пароль как отдельного пользователя, так и некоторой организации.

Критериями оценки эргономичности программного обеспечения является:

- цена ошибки – стоимость ошибки, произошедшей в результате некорректных или ошибочных действий пользователя;
- интуитивность графического интерфейса – совпадение между изображением в интерфейсе и ожидаемым действием;
- сложность обучения.

Цена ошибки оценивается с помощью определения количества возникающих ошибок и их стоимостью (например, в результате ошибочных действий пользователь удалил логин и пароль от корпоративной сети, после чего потерял к ней доступ. Для исправления данных ему потребуется пройти процедуру восстановления, утвержденную на его предприятии, что может занять некоторое время.) Стоимость данной ошибки можно оценить как среднюю – т. к.

данные возможно восстановить, но на это потребуется время). В отношении приложения «Менеджер паролей» выделим три стоимости ошибок: низкая (данные можно восстановить средствами самого приложения); средняя (данные возможно восстановить силами пользователя, с затратами времени); высокая (данные были безвозвратно утеряны).

Интуитивность графического интерфейса напрямую связанна с таким параметром как скорость обучения. Графический интерфейс не должен вводить пользователя в заблуждение. Значки и символы, используемые в приложении, должны соответствовать предоставляемому функционалу («знак» корзина – действие удалить). Также необходимо учитывать следующие показатели:

а) Наличие одинакового структурирования пользовательского интерфейса, настроек, документации и отчетов приложения.

б) Отсутствие дублирования функций, настроек, программных окон и элементов управления в разных компонентах приложения.

в) Наличие в пользовательском интерфейсе в каждый момент времени информации о результатах действий пользователя, реакции приложения на действия пользователя и состоянии приложения.

г) Наличие симметричности в программных элементах, выполняющих однотипные функции.

д) Отсутствие нефункциональных программных окно, «битых» ссылок и некорректной технической терминологии.

5.1.2.1 Окно входа в приложение

Окно изображено на рисунке 29, в хода состоит из нескольких элементов: кнопка аккаунты, выпадающее меню в нижнем правом углу.

По нажатию на кнопку «войти», открывается список, который содержит аккаунты, добавляемые пользователем, и является компонентов пользовательского интерфейса, который обеспечивает представление элементов. [1] Каждый элемент списка содержит в себе знак входа, выполненный в стиле Material

Design, по нажатию на который происходит вход в советующий значку аккаунт.

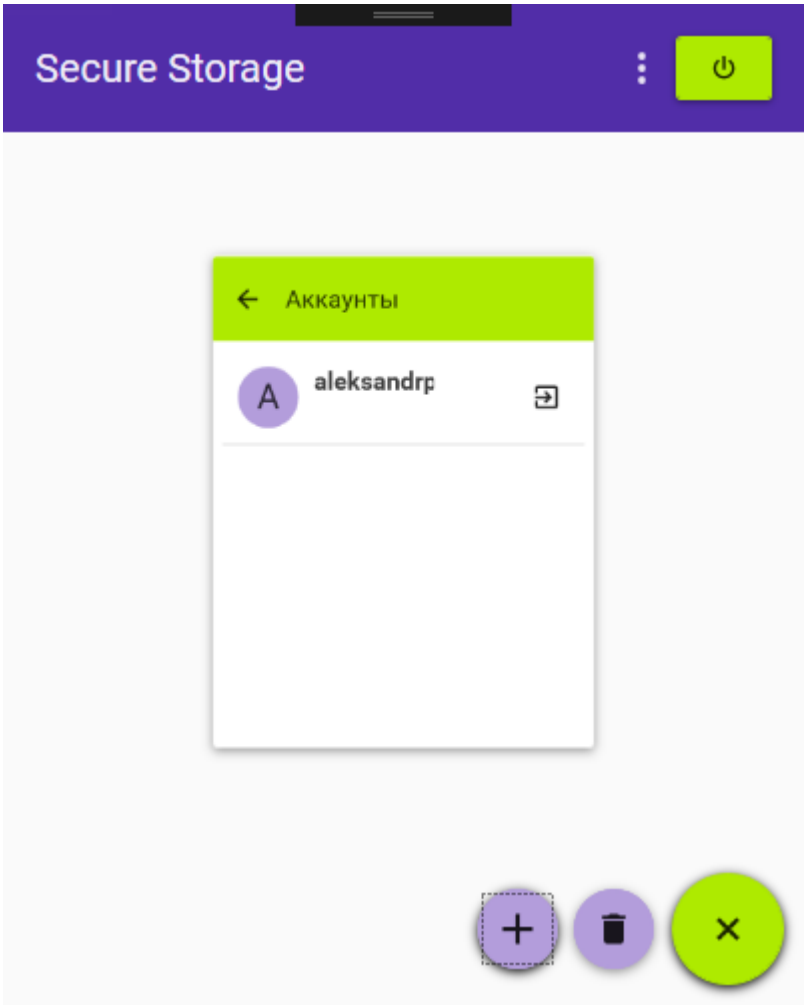

Рисунок 29 – Окно входа в приложение менеджер паролей

Действия добавления и удаления аккаунтов приложения расположены в раскрывающемся меню в правом нижем углу и обозначены соответствующими знаками. Для выбора аккаунтов, которые необходимо удалить, требуется нажать на миниатюру аккаунтов (она выполняет роль элемента типа CheckBox) после чего миниатюра заменится знаком «галочка» и затем выбрать опцию удалить.

В случае возникновения ошибки в результате каких-либо действий, пользователю показывается всплывающее меню, с описание ошибки и рекомендациями к дальнейшим действиям.

Для уменьшения стоимости ошибочного удаления аккаунта в данном

окне (удаление аккаунта, являющегося корневым элементом, в реляционной базе данных, приведет к удалению всех зависимых данных) реализовано мягкое удаление – т.е. удаляются только локально сохраненные данные, вся информация об аккаунте остается на сервере.

Таблица 13 – Анализ «окна входа» приложения «Менеджер паролей».

Критерий	Оценка	Пояснение
Цена ошибки	низкая	Предусмотренные механизмы «мягкого удаления», не позволят на данном этапе работы в приложении уничтожить данные пользователя по случайности, в результате ошибки или в результате злого умысла третьих лиц.
Интуитивность графического интерфейса	интуитивен	В интерфейсе используются только стандартные функциональные элементы (кнопка, выпадающее меню, список с выбором). Данные элементы используются пользователями ЭВМ повсеместно, во многих приложениях. Также все элементы графического интерфейса сопровождаются соответствующими значками (корзина – удалить, вход – авторизоваться). Все элементы дизайна выполнены в стиле Material Design с использованием общей цветовой схемы.
Сложность обучения	Низкая	Среднестатистический пользователь ЭВМ сталкивается с подобными интерфейсами, в повседневной жизни, как на ЭВМ, так и на мобильных устройствах. В случае ошибки, пользователю выдаётся предупреждение с инструкцией к дальнейшим действиям

Для полного удаления аккаунта пользователю необходимо авторизоваться, после чего пройти в раздел «настройки» -> «удаление аккаунта», после

чего еще раз ввести мастер-ключ. Формализуем всё вышесказанное в таблицу 13.

1.1.2.2 Основной интерфейс приложения «Менеджер паролей»

Основной интерфейс приложения представлен на рисунке 30 и состоит из: бокового меню, содержащего элементы для переключения между страницами приложения (страница менеджер паролей, страница настройки и т. д.) Весь интерфейс также соответствует цветовой и функциональной составляющей Material Design. Кнопки действий реализованы в аналогичном «окну входа» боковому меню и расположены на тех же местах. Соблюдается преемственность интерфейса.

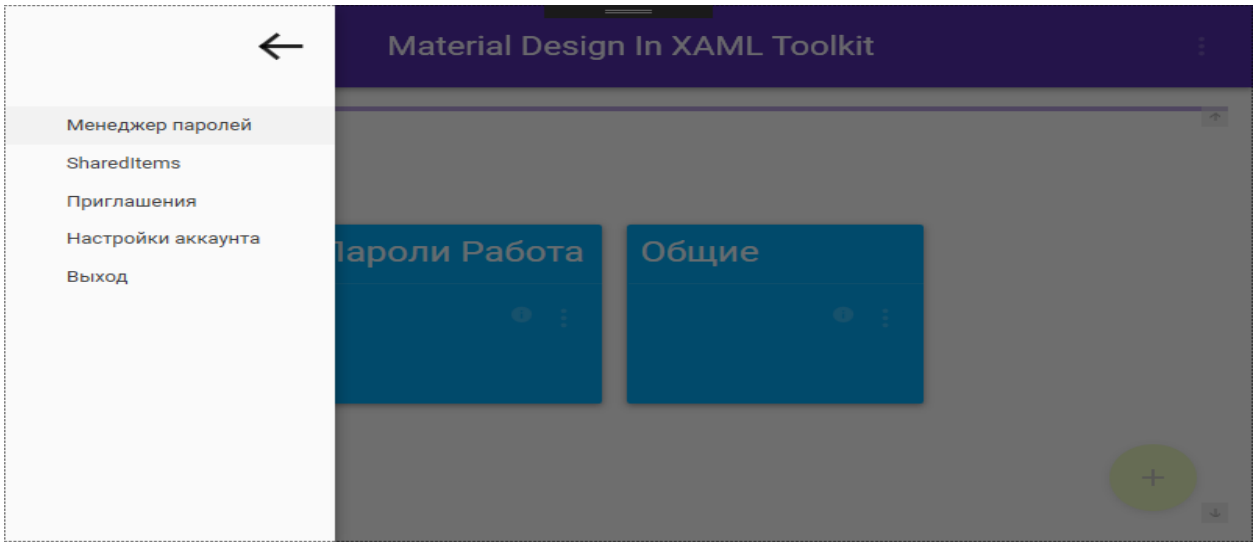

Рисунок 30 – Основной интерфейс приложения «Менеджер паролей»

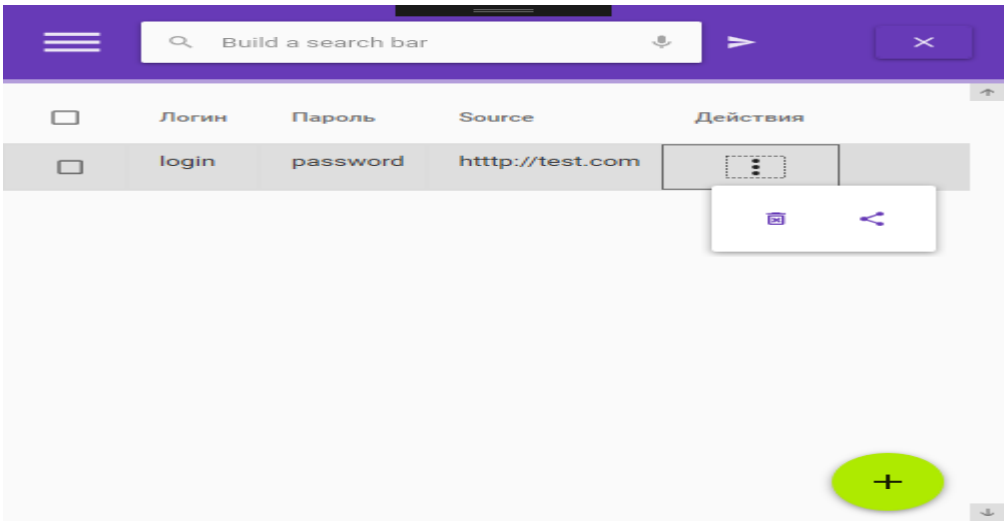

Рисунок 31 – Интерфейс управления паролями в группе

Действия, которые способны нанести ущерб пользователю в случае ошибочности (удаление аккаунта, данных), запрашивают подтверждение (рисунок 32) [2]. Ошибки в результате обмена с сервером или иные ошибки, выводятся пользователю с рекомендациями по дальнейшим действиям. Результат анализа представлен в таблице 14.

Таблица 14 – Результат анализа основного интерфейса приложения «Менеджер паролей».

Критерий	Оценка	Пояснение
Цена ошибки	низкая	Предусмотренные механизмы «мягкого удаления», не позволят на данном этапе работы в приложении уничтожить данные пользователя по случайности, в результате ошибки или в результате злого умысла третьих лиц.
Интуитивность графического интерфейса	интуитивен	В интерфейсе используются только стандартные функциональные элементы (кнопка, выпадающее меню, список с выбором). Данные элементы используются пользователями ЭВМ повсеместно, во многих приложениях. Также все элементы графического интерфейса сопровождаются соответствующими значками (корзина – удалить, вход – авторизоваться). Все элементы дизайна выполнены в стиле Material Design с использованием общей цветовой схемы.
Сложность обучения	Низкая	Среднестатистический пользователь ЭВМ сталкивается с подобными интерфейсами, в повседневной жизни, как на ЭВМ, так и на мобильных устройствах. В случае ошибки, пользователю выдаётся предупреждение с инструкцией к дальнейшим действиям

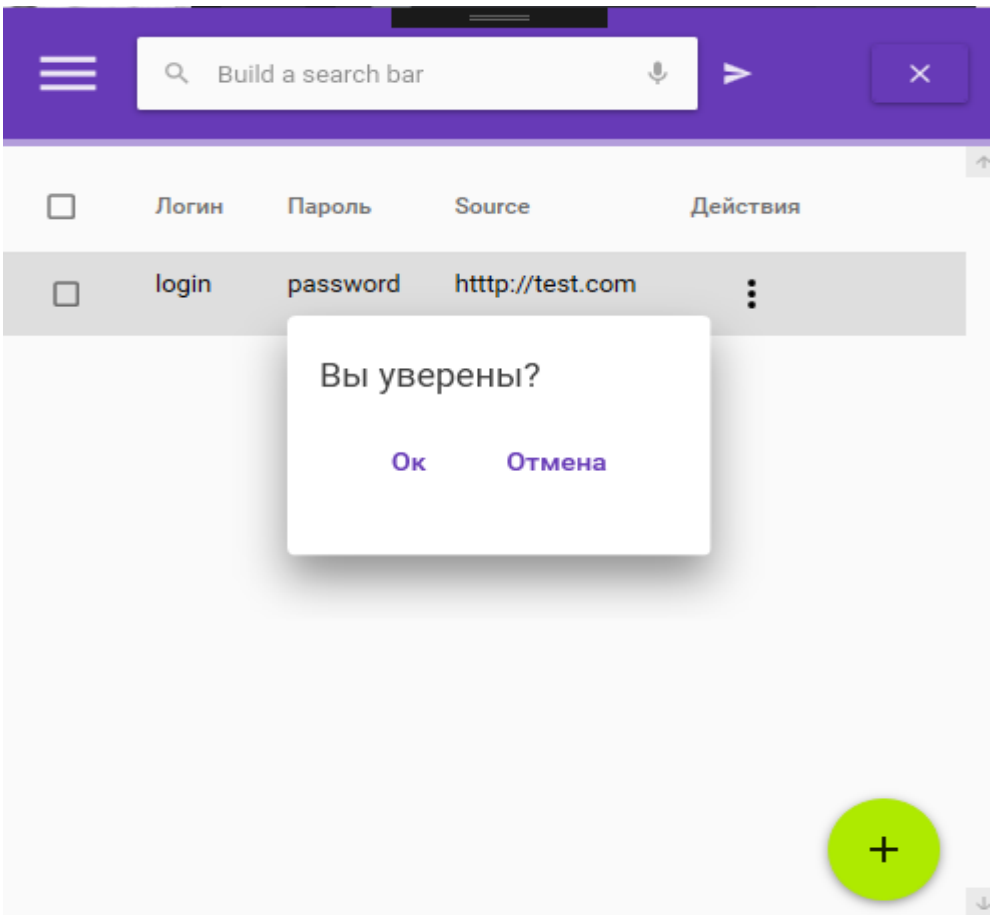

Рисунок 32 – Запрос подтверждения действия удаления записи пользователя

5.1.3 Анализ опасных и вредных факторов на рабочем месте пользователя ЭВМ

Программа «Менеджер паролей» поставляется в версия рассчитанных на использование как с ЭВМ (версия для персональных компьютеров), так и в WEB-версии. Следовательно, необходимо провести анализ рабочего места пользователя с ЭВМ.

Согласно СанПиН 2.2.2/2.4.1340-03, пункт 2, ПЭВМ должны соответствовать требованиям настоящих санитарных правил, и каждый их тип подлежит санитарно-эпидемиологической экспертизе с оценкой в испытательных лабораториях, аккредитованных в установленном порядке.[2]

Для ЭВМ установлены следующие нормы:

- уровни электромагнитных полей;
- акустический шум;
- визуальные показатели УОИ;

- мягкое рентгеновское освещение.

Так же в СанПиН приведены таблицы с подробным описанием норм.

Согласно пункту 6, СанПиН 2.2.2, рабочие столы следует размещать таким образом, чтобы видео дисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева. Так же следует ограничить отраженную блёскость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения мест по отношению к источникам естественного освещения.

Пункт 9, содержит требования к организации рабочих мест пользователей ПЭВМ. Согласно требованиям:

а) Расстояние между боковыми поверхностями должно быть не менее 1,2м.

б) Рабочие места при выполнении творческой работы требующего значительного умственного напряжения необходимо изолировать перегородками высотой 1,5-2,0м.

в) Экран монитора должен находиться на расстоянии 500-700 мм от глаз пользователя.

Пункт 10, в свою очередь содержит требования к организации рабочих мест с ПЭВМ для взрослых пользователей.

- высота рабочего стола в пределах 680-800 мм;
- глубина 800 или 1000 мм при нерегулируемой высоте;
- ширина – 800, 1000, 1200 и 1400 мм.

Пункты 11, 12 содержат рекомендации для пользователей общеобразовательных заведений и дошкольного возраста и, следовательно, не принимаются во внимание в текущей работе, т.к. в роли основных пользователей ИС выступают индивидуальные пользователи или же корпоративные пользователи различных информационных систем с парольным доступом.

Для возможных пользователей «Менеджера паролей», разработаем схему рабочего места в соответствии с перечисленными требованиями СаНПиН. Схема рабочего места, и источников освещения показана на рисунке 33.

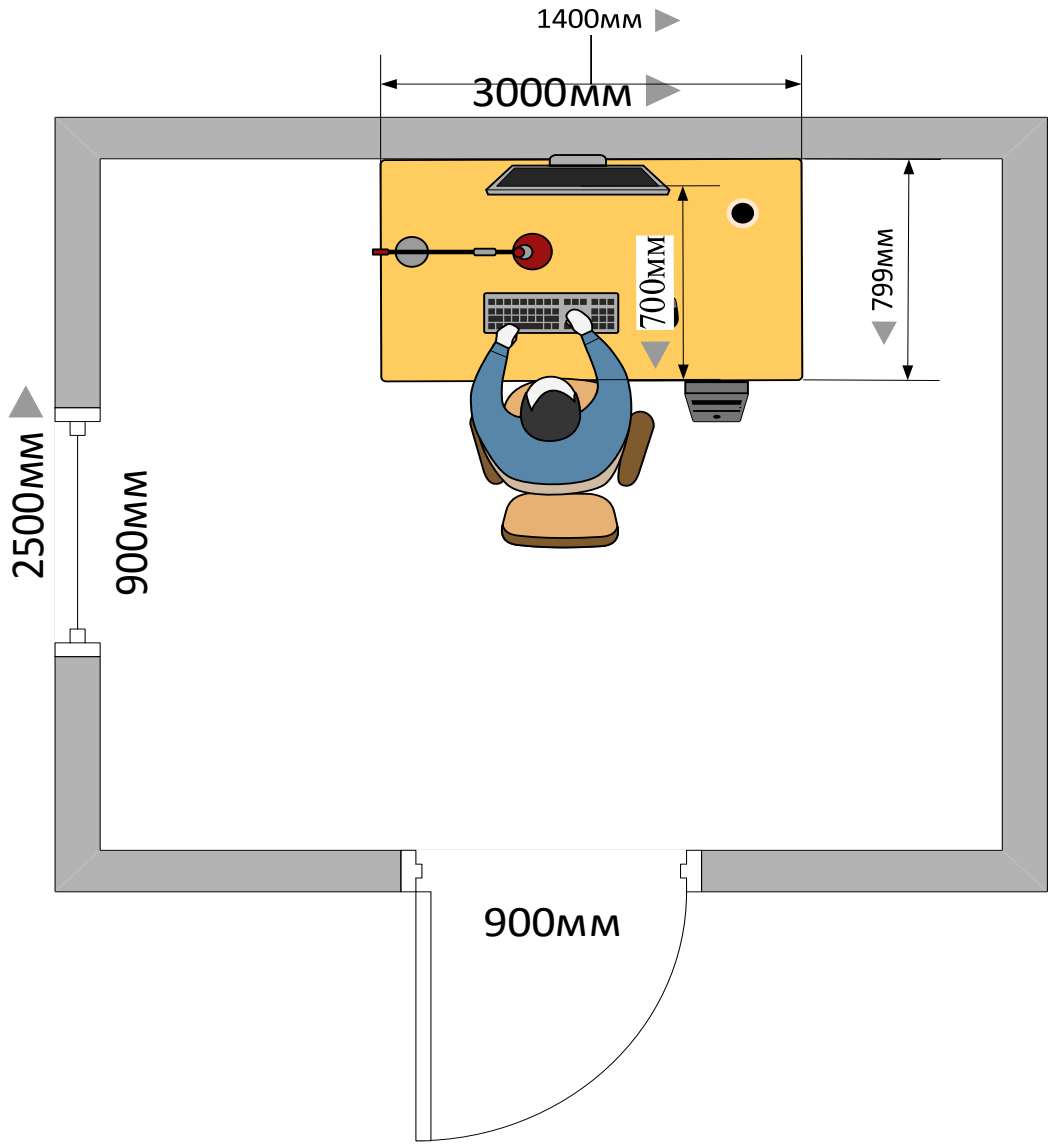

Рисунок 33 – Схема рабочего места

На схеме изображен человек работающий за ПЭВМ. Рабочее место оборудовано источником освещения, клавиатурой, мышью, принтером и удобной мебелью для отдыха.

5.2 Экологичность

Отходы производства – это остатки различного сырья, отработанных материалов и других продуктов, которые появляются в результате деятельности организации. Утилизация отходов – главная задача, которая стоит перед предприятием для обеспечения экологической безопасности окружающей среды. Организация и управление утилизацией отходов являются весьма сложным

делом. Заниматься этой деятельностью должны подготовленные специалисты, обладающие специальными знаниями.

Кроме того, что каждая организация обязано обеспечить уничтожение мусора, сама утилизация промышленных отходов должна отвечать необходимым стандартам и требованиям. Несоблюдение данных норм способно повлечь применение серьезных санкций к предприятию, вплоть до его закрытия.

Технологический процесс утилизации отходов зависит от утилизируемого материала. Например, утилизация строительных отходов значительно отличается от уничтожения пластика или химических веществ. Поэтому предприятия должны разрабатывать различные технологии утилизации и применять различное оборудование. Также значительно отличается утилизация твердых отходов от жидких.

Вопрос об экологической безопасности весьма важен, ведь от этого зависит здоровье людей, условия для нормального существования животных и растений.

Организация, в составе которой может применяться разрабатываемый продукт, может производить следующие виды отходов (по ГОСТ 30772-2001):

- Вторичная продукция [3] – материалы, комплектующие изделия, детали, функциональные узлы, блоки, агрегаты от различных объектов, утратившие свои потребительские свойства и непригодные для дальнейшей эксплуатации в соответствии с директивными требованиями и/или нормативной документацией, но представляющие собой товарную продукцию.
- Отходы производства [3] – остатки сырья, материалов, веществ, изделий, предметов, образовавшиеся в процессе производства продукции, выполнения работ (услуг) и утратившие полностью или частично исходные потребительские свойства.

Ко вторичной продукции, в организации использующей ЭВМ можно отнести: комплектующие и периферию ЭВМ. Существует несколько вариантов утилизации подобный отходов.

Первый вариант – это переработка объектов, состоящих преимущественно из пластмассы, однако для многих позиций микроэлектроники, таких как процессор, печатные платы, где полная переработка не возможна, используют частичную переработку в результате которых часть веществ утилизируется а другая часть отправляется на вторичное использование (например: некоторый редкоземельные металлы в составе печатных плат и микросхем).

Второй вариант – это продажа вторичной продукции как товаров бывших в употреблении. В этом случае организации нет необходимость оплачивать утилизацию/переработку отходов, а также появляется возможность частично окупить новое оборудование. Подобный способ используется ведущими дата-центрами (Reg.ru) (При обновлении серверов, компания продает устаревшие товары на вторичном рынке).

Отходы производства в свою возможно только утилизировать. Для этого отходы сортируются по типу (стекло, пластик, металлы и т. д.), после чего отправляется в центры утилизации.

5.3 Безопасность при возникновении чрезвычайных ситуаций

Одной из частых чрезвычайных ситуаций в организациях – пожар. Основные причины пожаров на предприятиях - неосторожное обращение с огнем, оставленные без присмотра электроприборы, проведение с нарушениями требований правил пожарной безопасности огневых, строительных и других пожароопасных работ, курение в не установленных местах, использование легко-воспламеняемых веществ т. д.

Согласно НПБ 105-03 помещения с ЭВМ относятся к пожароопасным помещениям категорий В1-В4 (т. к. содержат материалы способные при взаимодействии с водой или друг с другом гореть). Мебель и другие бытовые предметы не должны препятствовать эвакуации, а также все провода должны быть спрятаны в стену или кабель-каналы.

Для предотвращения возможного пожара необходимо соблюдать следующие правила:

- не хранить и не применять горючие жидкости, взрывчатые вещества, баллоны с газами и рядом с ЭВМ;
- не использовать электронагревательные приборы;
- не эксплуатировать провода электроприборов с поврежденной изоляцией;
- не пользоваться поврежденными розетками, и прочим электрооборудованием;
- не накрывать светильники, бытовые приборы бумагой, тканью и другими горючими материалами;
- не курить в помещении;
- оставлять без наблюдения включенную в сеть радиоэлектронную ПЭВМ;
- не пользоваться неисправной аппаратурой;
- не разрешается ремонтировать блоки ЭВМ непосредственно в помещениях, где они располагаются;
- не нарушать правила эксплуатации ПЭВМ;
- раз в 3 месяца необходимо проводить санитарную очистку;

По окончании работы необходимо обесточить все электроприборы и осмотреть помещения на наличие признаков возгорания, а также необходимо выключить автомат питания в распределительном щите, если такой имеется.

Если же всё-таки случилось возгорание, необходимо позвонить в пожарную службу, сообщить всю необходимую информацию, подготовить к эвакуации материальные ценности, документацию и покинуть здание через запасные выходы. Если нет возможности покинуть здание, то необходимо закрыться в менее задымлённой комнате, не дать дыму попадать в комнате любыми подручными средствами и открыв все окна ожидать помощи спасательной бригады.

5.4 Комплекс физических упражнений при работе за ЭМВ

В настоящее время люди огромное количество времени посвящают компьютеру, что приводит к высокой нагрузке как на опорно-двигательный, так и на зрительный аппараты. При незначительных изменениях проблема может быть незаметна, однако постепенно формируются серьезные нарушения, такие

как: снижения зрения, искривление осанки, артриты. В таком случае рекомендуется выполнять комплекс упражнений для снятия усталости за компьютером. Благодаря регулярным тренировкам можно предотвратить появление многих проблем.

Упражнения для улучшения кровообращения в мозговой области.

а) Исходное положение на стуле, руки свесить, расслабиться. Медленно наклоните голову назад. Считаем до трех, медленно. Затем занимаем исходное положение. Затем медленно наклоняем голову вперед. Считаем до трех и возвращаемся в исходное положение.

б) Исходное положение, сидя на стуле, руки на поясе. Делаем все как в первом упражнении, но голову наклоняем сначала к левому плечу, потом к правому.

в) Можно сидя или стоя. Левую руку заносим за голову и тянемся к правому плечу, поворачиваем голову на лево. Считаем до трех и проделываем все то же самое, но с правой рукой и голову поворачиваем на право.

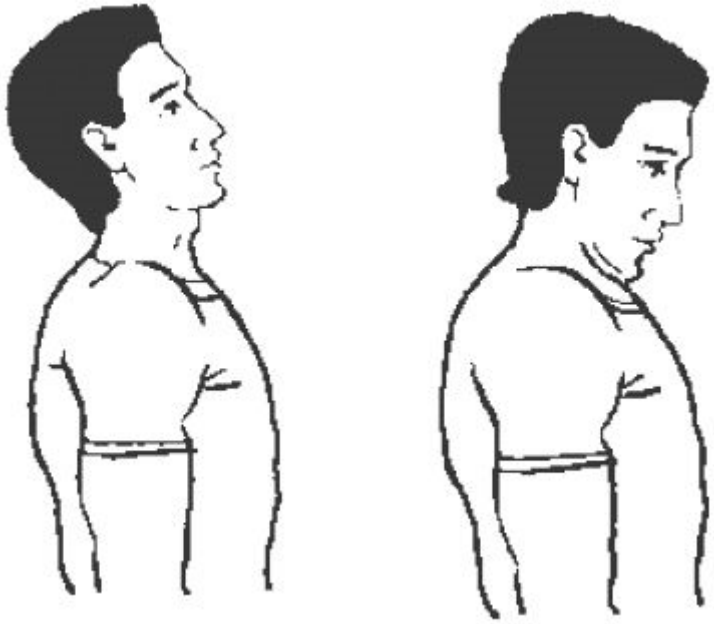

Рисунок 34 – Иллюстрация упражнений для шейного отдела

Упражнения для снятия усталости с кистей рук и плечевого пояса.

а) Упражнение можно выполнять как сидя, так и стоя. Левую руку вытягиваем вперед, правую поднимаем вверх. Меняем положения рук, поочередно. Темп выполнения средний.

б) Положение стоя. Руки тыльной стороной кисти прижать к поясу. Свести локти вместе голову наклонить вперед. Локти развести в стороны и пытаться свести за спиной, голову соответственно наклонить назад.

в) Выполнять сидя на стуле. Поднять руки вверх сжимать и разжимать поочередно кисти рук.

Упражнения для снятия напряжения с туловища.

а) Исходное положение, стоя, руки за голову, ноги чуть шире плеч. Поворачивать таз влево и вправо. Плечевой пояс должен быть неподвижен.

б) Положение аналогично первому упражнению. Тазом делаем круговые вращения почасовой стрелки и против часовой стрелки, поочередно.

в) Стойка – ноги врозь. Наклоняемся вперед, правая рука скользит по ногам вниз, а левая поднимается вдоль тела. Далее проделываем то же самое, но меняем положение рук.

ЗАКЛЮЧЕНИЕ

В настоящее время, создается огромное количество различных сервисов, в связи с чем, возросла актуальность продуктов, решающих проблему управления аутентификационными парами.

При выполнении бакалаврской работы был проведен анализ существующих решений (таблицы 1-3), в результате которого, в трех исследуемых программах были найдены уязвимости к атаке «подмена .dll» и атаке, целью которой является получение данных из окна приложения. На основании данных анализа определили цели и функции программного продукта. После чего было выполнено проектирование структуры приложения и базы данных.

Итогом проектирования стало определение требований к продукту, структура и функционал программных модулей. На этапе разработки был произведен выбор методологии разработки программного обеспечения, в результате которого была выбрана практика «Экстремального программирования», т. к. она обладает значительными преимуществами при разработке современных объектно-ориентированных приложений.

В качестве средств реализации были выбраны:

- среда разработки Visual Studio 2019;
- язык программирования C#;
- для приложения-клиента платформа .NET Core 3.0;
- для серверного модуля фреймворк ASP .NET Core 2.2;

Также в рамках главы 3: разработаны модули приложения и отражена схема их взаимодействия (рисунок 17); рассмотрены трудности и их решение при реальной эксплуатации приложения-сервера на базе платформы ASP .NET Core 2.2; приведено описание экранных форм.

Для разработанного приложения были изучены модули угроз и варианты их реализации нарушителем, а также меры по их предотвращению; установлены рекомендации по безопасности и экологичности.

Разработанное кроссплатформенное клиент-серверное приложение предлагает пользователю систему для хранения и управления данными типа аутентификационная пара, а также возможность поделиться данными с другими пользователями системы.

На текущий момент программный продукт находится на этапе тестирования и проходит процедуру официальной регистрации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Информационные системы: Учебное пособие. / Е. В. Бурцева [и др.]. – Тамбов: ТГТУ, 2009. – 128 с.

2 НПБ 105-03. Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности

3 Объектно-ориентированное программирование: учебное пособие для прикладного бакалавриата / А. Ф. Тузовский. – Москва: Издательство Юрайт, 2019. – 206 с. – (Университеты России). – ISBN 978-5-534-00849-4. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: https://biblio-online.ru/bcode/434045 (дата обращения: 31.05.2019).

4 Принципы, паттерны и методики гибкой разработки на языке C#. / Пер. с англ. – СПб.: Символ-Плюс, 2011. – 768 с.

5 Чистый код: создание, анализ и рефакторинг. Библиотека программиста. / Пер. с англ. – СПб.: Питер, 2010. – 464 с.

6 CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. – СПб.: Питер, 2013. – 896 с.

7 CodinGame: Creating Web API in ASP .NET Core 2.0 [Электронный ресурс]. URL: https://www.codingame.com/playgrounds/35462/creating-web-api-in-asp-net-core-2-0/part-1---web-api

8 DigitalOcean: Initial Server Setup with Ubuntu 18.04 [Электронный ресурс]. URL: https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-18-04

9 Habr: MVVM: полное понимание [Электронный ресурс]. URL: https://habr.com/ru/post/338518/

10 IntelliTect: Getting started with MVVM pattern using Windows Presentation Framework [Электронный ресурс]. URL: https://intellitect.com/getting-started-model-view-viewmodel-mvvm-pattern-using-windows-presentation-framework-wpf/

11 MaterialDesignXaml.net: Material design in WPF Guide [Электронный ресурс]. URL: http://materialdesigninxaml.net/
12 Medium.com: Setup Entity Framework Core [Электронный ресурс]. URL: https://medium.com/@balramchavan/setup-entity-framework-core-for-mysql-in-asp-net-core-2-5b40a5a3af94
13 Metanit.com: Паттерны проектирования в C# и .NET [Электронный ресурс]. URL: https://metanit.com/sharp/patterns/
14 Metanit.com: Полное руководство по языку программирования C# и платформе .NET 4.7 [Электронный ресурс]. URL: https://metanit.com/sharp/tutorial/
15 Microsoft Docs: Документация к ASP .Net Core [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.2
16 Microsoft Docs: Документация к System.Net.Http [Электронный ресурс]. URL: https://docs.microsoft.com/enus/dotnet/api/syытem.net.http.httpclent?view=net framework-4.7.2
17 Official Knowledge Base by phoenixNAP: How To Configure MySQL 8.0 On Ubuntu 18.04 [Электронный ресурс]. URL: https://phoenixnap.com/kb/how-to-install-mysql-on-ubuntu-18-04
18 Telerik.com: Build and Deploy Your ASP.NET Core Application with Apache [Электронный ресурс]. URL: https://www.telerik.com/blogs/build-deploy-asp-net-core-application-apache
19 Tools.ietf.org: Стандарт RFC7519 (JSON Web Token) [Электронный ресурс]. URL: https://tools.ietf.org/html/rfc7519

ПРИЛОЖЕНИЕ А

Диаграммы IDEF1X базы данных приложения

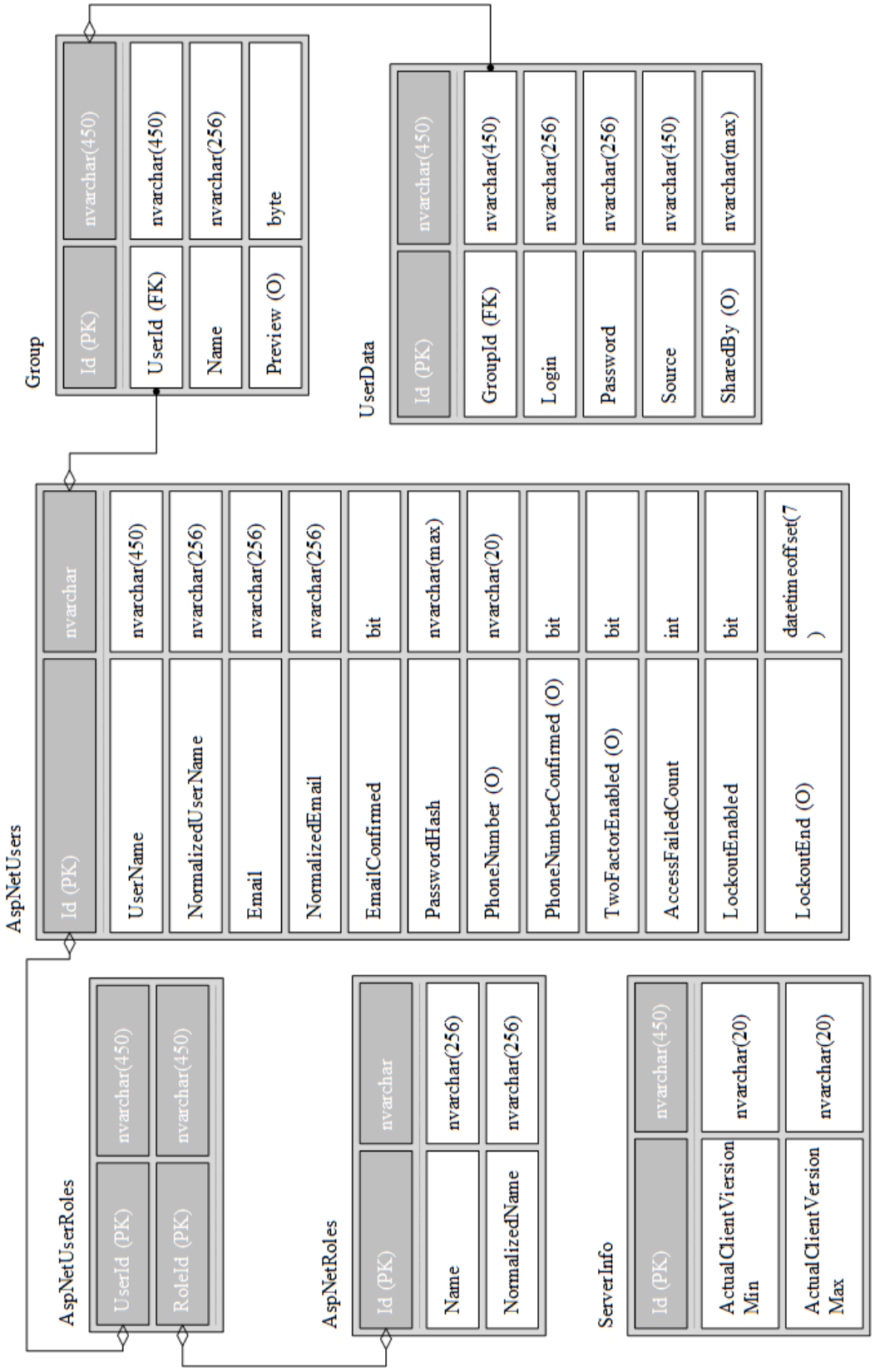

Рисунок А.1 – Диаграмма IDEF1X базы данных приложения

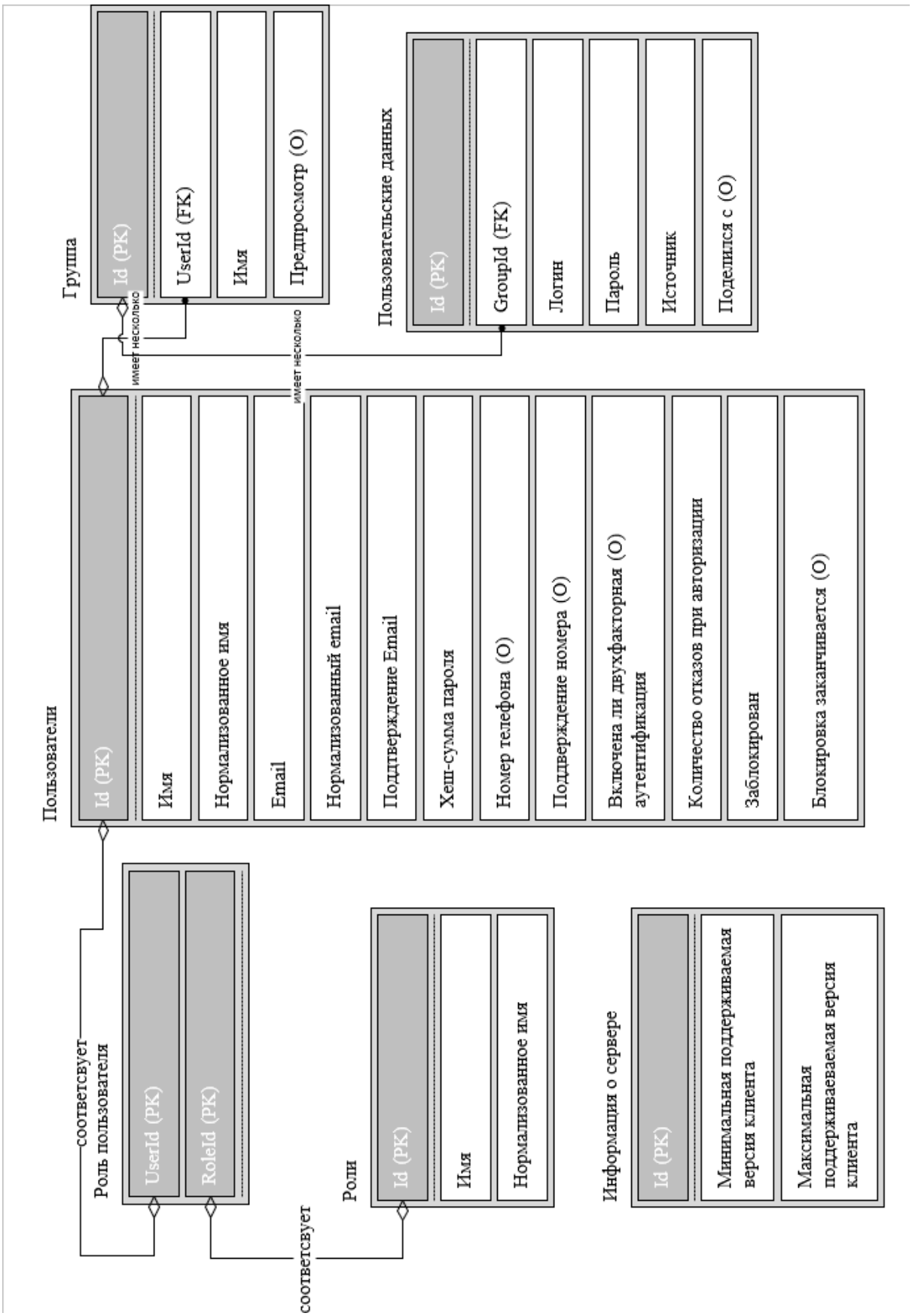

Рисунок А.2 –Логическая диаграмма базы данных приложения

ПРИЛОЖЕНИЕ Б

Настройка сервисов сервера аутентификации

```
ссылок: 0 | 0 исключения
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_2);
    services.AddCors();
    services.AddDbContext<ApplicationDbContext>(options =>
    {
        options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection"));
        options.UseOpenIddict();
    });

    services.AddIdentity<ApplicationUser, IdentityRole>()
        .AddEntityFrameworkStores<ApplicationDbContext>()
        .AddDefaultTokenProviders();

    services.Configure<IdentityOptions>(options =>
    {
        options.ClaimsIdentity.UserIdClaimType = OpenIdConnectConstants.Claims.Name;
        options.ClaimsIdentity.UserNameClaimType = OpenIdConnectConstants.Claims.Subject;
        options.ClaimsIdentity.RoleClaimType = OpenIdConnectConstants.Claims.Role;
    });

    services.AddOpenIddict()
        .AddCore(options =>
        {
            options.UseEntityFrameworkCore()
                   .UseDbContext<ApplicationDbContext>();
        })
        .AddServer(options =>
        {
            options.UseMvc();
            options.EnableTokenEndpoint("/auth_token");
            options.AllowPasswordFlow();
            options.AcceptAnonymousClients();
            //options.DisableHttpsRequirement();
        })
        .AddValidation();
}
```

Рисунок Б.1 – Листинг кода конфигурации сервера-авторизации

ПРИЛОЖЕНИЕ В

Создание JWT токена на стороне сервера

```
private async Task<AuthenticationTicket> CreateTicketAsync(OpenIdConnectRequest request, ApplicationUser user)
{
    var principal = await _signInManager.CreateUserPrincipalAsync(user);
    var ticket = GetConfiguratedTicket(principal, request.GetScopes());
    // Note: by default, claims are NOT automatically included in the access and identity tokens.
    // To allow OpenIddict to serialize them, you must attach them a destination, that specifies
    // whether they should be included in access tokens, in identity tokens or in both.
    foreach (var claim in ticket.Principal.Claims)
    {
        if (claim.Type == SecurityStamp)
            continue;
        SetClaimDestinations(claim, ref ticket);
    }
    return ticket;
}
ссылка: 1 | 0 исключения
private void SetClaimDestinations(Claim claim, ref AuthenticationTicket ticket)
{
    var destinations = new List<string>
        {
            OpenIdConnectConstants.Destinations.AccessToken
        };

    // Only add the iterated claim to the id_token if the corresponding scope was granted to the client application.
    // The other claims will only be added to the access_token, which is encrypted when using the default format.
    if ((claim.Type == Name && ticket.HasScope(OpenIdConnectConstants.Scopes.Profile)) ||
        (claim.Type == Email && ticket.HasScope(OpenIdConnectConstants.Scopes.Email)) ||
        (claim.Type == Role && ticket.HasScope(OpenIddictConstants.Claims.Roles)))
    {
        destinations.Add(OpenIdConnectConstants.Destinations.IdentityToken);
    }

    claim.SetDestinations(destinations);
}
```

Рисунок В.1 – Листинг кода создания токенов

ПРИЛОЖЕНИЕ Г

Декомпозиции логики приложения-клиента и логики приложения-сервера

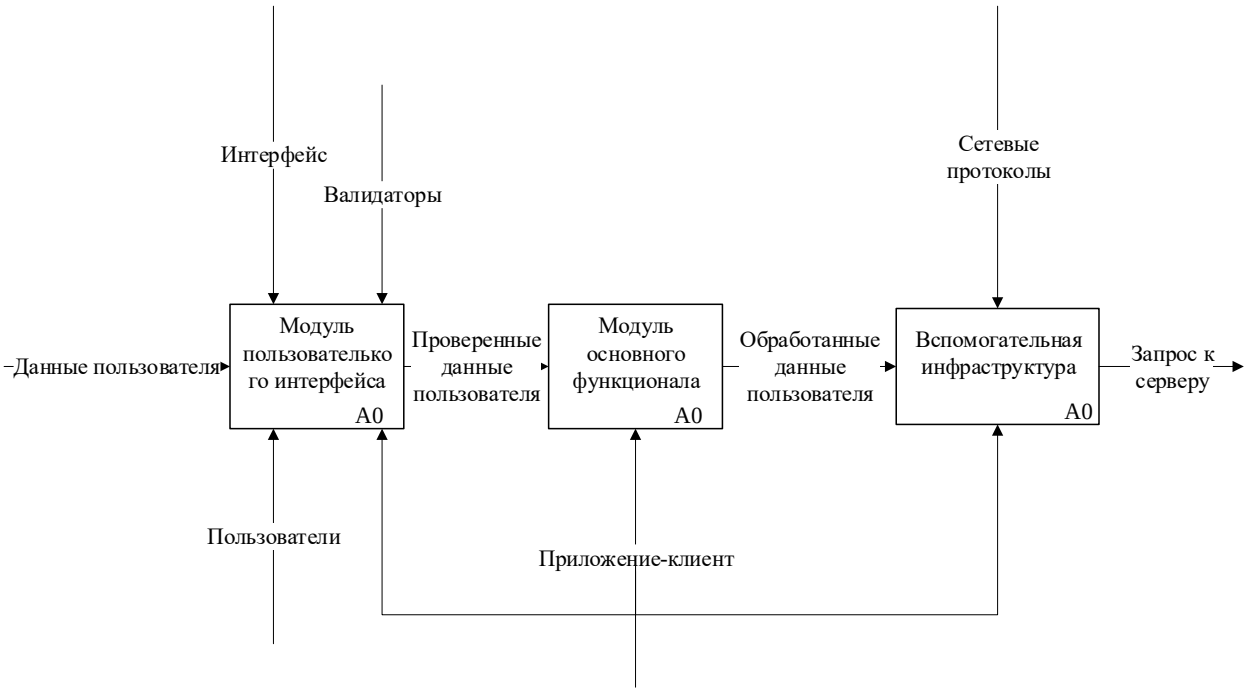

Рисунок Г1 – Декомпозиция логики приложения-клиента

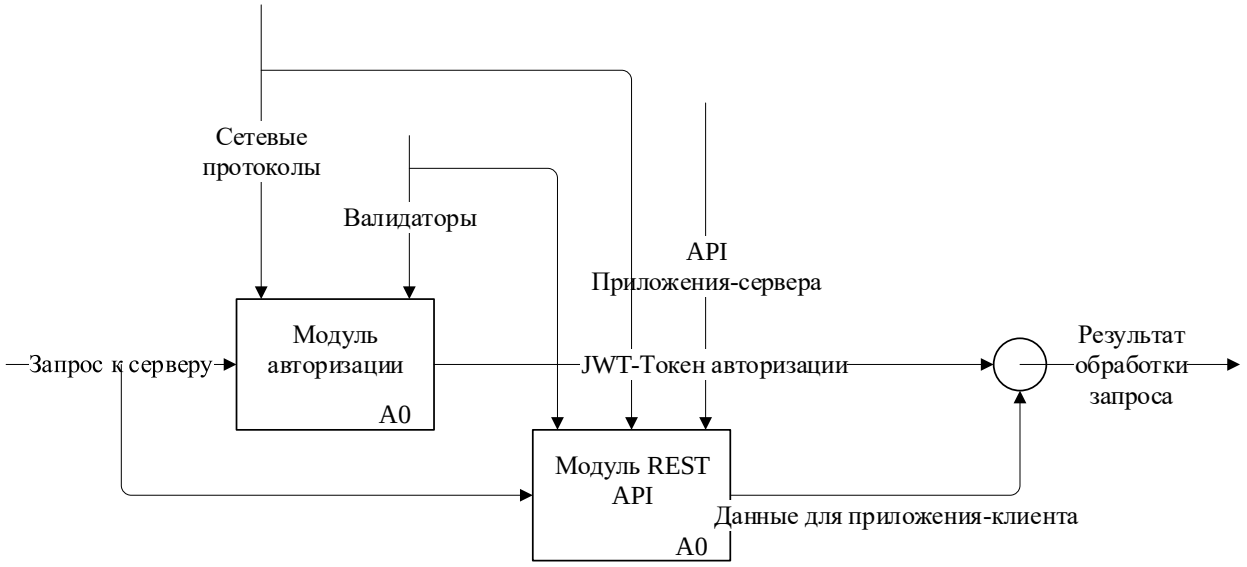

Рисунок Г2 – Декомпозиция логики приложения-сервера

ПРИЛОЖЕНИЕ Д

Техническое задание

Работа выполняется в рамках выпускной квалификационной работы Разработка кроссплатформенного клиент-серверного приложения «Менеджер паролей»

Основания для разработки

Основанием для разработки служит задание на выпускной проект

Полное наименование информационной системы:

«Менеджер паролей».

Условное обозначение системы:

«SUDa»

Исполнитель: Понизов А.В.

Назначение разработки

Информационная система «SUDa» предназначена для создание комфортной среды по управлению данными пользователя типа аутентификационная пара.

Цель информационной системы:

Управление аутентификационными парами (логин/пароль).

Технические требования

3.1 Требования, предъявляемые к информационной системе.

Структура информационной системы:

– Модель REST API – позволяет использовать, вместе с сервером-приложения, приложения-клиенты для любых платформ;

– Модульная структура. Система состоит из максимально независимых компонентов, имеющих собственные поведение и состояние.

Способы доступа к системе:

Доступ к системе определяется штатными средствами вычислительной системы. К штатным средствам относится совокупность аппаратных средств вычислительной техники и программного обеспечения.

Совместимость системы:

Вычислительная система должна соответствовать современным стандартам качества, пожарной и электробезопасности, и гигиенических требований.

Диагностирование системы:

Диагностирование (а также восстановление в случае технических сбоев) системы осуществляется с привлечением внешних технических специалистов, осуществляющих обслуживание программного обеспечения и средств вычислительной техники.

3.2 Требования к функциональным характеристикам.

Состав выполняемых функций. Перечень функций разрабатываемого программного обеспечения:

регистрация в системе;

авторизация в системе;

создание и удаление групп для данных;

поиск по данным;

возможность поделиться данными с другими пользователями системы

Перечень используемых подсистем:

– Подсистема Пользователь – специалист, обеспечивающий ввод данных и получение результата;

– Подсистема Интерфейс – интерфейс взаимодействия между пользователями системы и прочими компонентами системы. Осуществляет логический контроль вводимых данных, предоставляет отчеты в удобной для человека форме;

Показатели назначения:

– возможность использовать разные улично-дорожные сети и их фрагменты;

– однопользовательская работа, возможно внедрение многопользовательской работы;

– возможность использовать различные параметры модели;

– время формирования отчетов – определяется составом и количеством анализируемых данных;

– масштабируемость системы – не обеспечивается, не требуется;

3.3 Требования к надежности

Информационная система должна сохранять работоспособность при возникновении следующих ситуаций:

– сбой в системе электроснабжения, повлекший перезагрузку аппаратного или программного обеспечения;

– ошибки в работе аппаратных средств, восстановление работы должно осуществляться средствами операционной системы;

– ошибки в системном программной обеспечении, восстановление работы должно осуществляться средствами операционной системы.

Аппаратное обеспечение информационной системы должно иметь средства защиты от бросков напряжений и короткого замыкания.

3.4 Требования к безопасности

Аппаратное обеспечение должно иметь защитное заземление согласно ГОСТ 12.1.030-81 и ПУЭ.

Требования пожарной безопасности определяются нормам безопасности на бытовое электрооборудование и зданий, в которых будут располагаться компоненты информационной системы.

Факторы, могущие оказывать вредное воздействие на организм человека должны нормироваться в соответствии с СанПиН 2.2.2./2.4.1340-03.

3.5 Требования к эргономике

– работа должна осуществляться с использованием графического интерфейса пользователя;

– работа в информационной системе осуществляется с использованием стандартных устройств ввода (клавиатура, мышь) и стандартных устройств вывода (монитор, принтер);

– взаимодействие с информационной системой должно осуществляться на языке пользователей системы;

– все работники должны иметь офисную мебель, соответствующую требованиям занимаемыми ими должностей;

– все используемое оборудование должно иметь соответствующие сертификаты качества, гарантирующие основные эргономические характеристики.

3.6 Требования к патентной чистоте

Работа информационной системы и пользователей должна осуществляться в соответствии с лицензионными соглашениями на аппаратное и программное обеспечение.

3.7 Условия эксплуатации и требования к составу и параметрам технических средств.

Для работы системы должен быть выделен ответственный оператор. Требования к составу и параметрам технических средств уточняются на этапе эскизного проектирования системы.

3.8 Требования к информационной и программной совместимости

Программа должна работать на платформах Windows 10 и выше.

3.9 Требования к транспортировке и хранению

Программа поставляется на лазерном носителе информации. Программная документация поставляется в электронном и печатном виде.

3.10 Специальные требования

Программное обеспечение должно иметь дружественный интерфейс, рассчитанный на пользователя (в плане компьютерной грамотности) средней квалификации. Ввиду объемности проекта задачи предполагается решать поэтапно. При этом модули ПО, созданные в разнос время, должны предполагать возможность наращивания системы и быть совместимы друг с другом. Поэтому документация на принятое эксплуатационное ПО должна содержать полную информацию, необходимую для работы программистов с ним.

Язык программирования – по выбору исполнителя, должен обеспечивать возможность интеграции программного обеспечения с некоторыми видами периферийного оборудования.

4 Требования к программной документации

Перечень регламентирующей документации:

– Обоснование по созданию информационной системы;

– Схема организационной структуры информационной системы;

– Руководство программиста;

– Техническое задание;

– Пояснительная записка проекта.

5 Порядок контроля и приемки

После передачи Исполнителем отдельного функционального модуля программы Заказчику, последний имеет право тестировать модуль в течение 7 дней. После тестирования Заказчик должен принять работу по данному этапу или в письменном виде изложить причину отказа от принятия. В случае обоснованного отказа Исполнитель обязуется доработать модуль.