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
«_____»_____2022 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка мобильного приложения «Менеджер паролей» с синхронизацией на компьютере

Выполнил студент группы 855-об	_____ (подпись, дата)	В.В. Евдокимова
Руководитель доцент, канд. техн. наук	_____ (подпись, дата)	Т.А. Галаган
Консультант по безопасности и экологичности доцент, канд. техн. наук	_____ (подпись, дата)	А.Б. Булгаков
Нормоконтроль инженер кафедры	_____ (подпись, дата)	В.Н. Адаменко

Благовещенск 2022

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

УТВЕРЖДАЮ
Зав. кафедрой
_____А.В. Бушманов
«_____»_____

ЗАДАНИЕ

К выпускной квалификационной работе студента Евдокимовой Виктории Вадимовны

1. Тема выпускной квалификационной работы: Разработка мобильного приложения «Менеджер паролей» с синхронизацией на компьютере
(утверждена приказом от 05.04.2022 №679-уч)

2. Срок сдачи студентом законченной работы 21.06.2022 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): принципы работы и особенности работы программ менеджеров паролей, проектирование приложения, программная реализация, обеспечение информационной безопасности, безопасности и экологичность

5. Перечень материалов приложения: техническое задание на разработку программного обеспечения, шифрование и дешифрование данных, хэширование пароля и проверка на подлинность, свидетельство о регистрации программы

6. Консультанты по выпускной квалификационной работе
по безопасности и экологичности – Булгаков А.Б., доцент, кандидат технических наук

7. Дата выдачи задания 07.02.2022 г.

Руководитель выпускной квалификационной работы: Галаган Татьяна Алексеевна, доцент, кандидат технических наук
(фамилия, имя, отчество, должность, ученая степень, ученое звание)

Задание принял к исполнению _____
(подпись студента)

РЕФЕРАТ

Выпускная квалификационная работа содержит 71 с, 33 рисунка, 9 таблиц, 4 приложение, 20 источников, 8 нормативных ссылок.

АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ, ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА, КЛИЕНТ-СЕРВЕРНАЯ АРХИТЕКТУРА.

В работе произведены проектирование и разработка мобильного приложения «Менеджер паролей» с синхронизацией на компьютере.

Цель работы: проектирование и разработка приложения «Менеджер паролей», в функционал которого входит генерация паролей с отображением времени его жизни. Помимо стандартных функций, приложение оснащено возможностью синхронизации в локальной сети с приложением на компьютере с применением шифрования. Данное приложение разработано для облегчения хранения паролей и для создания надежного пароля.

В ходе работы были выполнены следующие задачи:

− анализ существующих решений;

− проектирование баз данных и функциональных модулей;

− разработка программного продукта на основании выбранных архитектуры и средств разработки;

− оценка информационной безопасности и безопасности жизнедеятельности.

Результатом выпускной квалификационной работы является полностью готовое мобильное приложение «Менеджер паролей» с синхронизацией на компьютере.

НОРМАТИВНЫЕ ССЫЛКИ

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

ГОСТ Р ИСО/МЭК 27002-2021. Информационная технология. Методы и средства обеспечения безопасности. Свод норм и правил применения мер обеспечения информационной безопасности.

ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению.

СП 2.2.3670-20. Санитарно-эпидемиологические требования к условиям труда.

СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания.

СП 52.13330.2016. Естественное и искусственное освещение.

ГОСТ 30772-2001. Ресурсосбережение. Обращение с отходами. Термины и определения.

ГОСТ Р 51275-2006. Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Общие положения.

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

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ОС	операционная система
ПО	программное обеспечение
ПЭВМ	персональная электронно-вычислительная машина
ФИО	фамилия, имя, отчество
ФСТЭК	Федеральная служба по техническому и экспортному контролю
ЭВМ	электронно-вычислительная машина
AES	Advanced Encryption Standard (расширенный стандарт шифрования)
API	Application Programming Interface (программный интерфейс приложения)
HTTP	HyperText Transfer Protocol (протокол для передачи гипертекста)
HTTPS	HyperText Transfer Protocol Secure (безопасный протокол передачи гипертекста)
IP-адрес	Internet Protocol (протокол интернета)
MVC	Model-View-Controller (модель-представление-контроллер)
MVVM	Model-View-ViewModel (модель-представление-модель представления)
SMS	Short Message Service (служба коротких сообщений)
USB	Universal Serial Bus (универсальная последовательная шина)
VPN	Virtual Private Network (виртуальная частная сеть)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В настоящее время существует большое количество сервисов, которые необходимы пользователям разной сферы деятельности и увлечения. Для каждого сервиса необходимо помнить логин и пароль, которые исчисляются в десятках. Появляется вопрос о хранении всех аутентификационных данных. Многие пренебрегают безопасностью этой важной информации, из-за чего те могут быть неизбежно утеряны или украдены злоумышленником. Необходимо безопасное, но в то же время легкодоступное для самого пользователя место, с помощью которого он сможет иметь доступ к сервисам как на мобильном устройстве, так и на домашнем компьютере.

В ходе данной работы будет рассмотрена не только разработка мобильного приложения, но и осуществление безопасности данных. Так как аутентификационные данные предоставляют доступ к некоторым конфиденциальным данным, эти данные нужно хранить в надлежащем виде (использовать хеширование с солью, шифровать данные в базах данных). Помимо этого, необходимо регистрировать мобильное устройство, с которого производится вход в систему, и использовать физическую защиту для него (ГОСТ Р ИСО/МЭК 27002-2021). В качестве последнего выступает как пароль на самом устройстве, так и дополнительный пароль в мобильном приложении.

Но защиты мобильного устройства недостаточно, так как все пароли хранятся на сервере. Злоумышленник, способный получить доступ к серверу, обойдя защитные меры, сможет обратиться ко всем аутентификационным данным, а следовательно, и к конфиденциальным данным (паспортные данные, банковские карты, электронная почта и тд.) тех или иных сервисов.

Тогда необходимо такое место резервного хранения, которым будет владеть сам пользователь приложения. В качестве такого может выступать локальный сервер, который расположен на компьютере пользователя. Таким образом, для обеспечения безопасности используется также локальная сеть, которая не имеет дополнительных посредников-злоумышленников.

Подытожив вышесказанное, стоит отметить, что проблемы хранение и защита таких данных, как логин и пароль, актуальны на сегодняшний день. Далее рассмотрим возможное решение данных проблем.

1 ПРИНЦИПЫ РАБОТЫ И ОСОБЕННОСТИ РЕАЛИЗАЦИИ ПРОГРАММ МЕНЕДЖЕРОВ ПАРОЛЕЙ

В рамках темы разработки мобильного менеджера паролей рассмотрим, что из себя представляет данное ПО.

Менеджер паролей – программное обеспечение, которое хранит пароли пользователей от различных сервисов. Местом хранения данных в таких программах является локальная или глобальная база данных, а конфиденциальная информация представляется там в зашифрованном виде.

Дополнительной функцией у некоторых менеджеров является заполнение электронных форм. Для реализации такой функции приложению необходимо иметь данные о пользователе (ФИО, дата рождения, паспортные данных и тд). В таком случае, программное обеспечение должно обладать достаточной защитой для хранения персональных данных пользователя.

Некоторые менеджеры обладают функцией аутентификации, что позволяют пользователю не авторизоваться в поддельном сайте того или иного сервиса, тем самым защитив его конфиденциальные и персональные данные.

Реализации парольных менеджеров:

− браузерные (сетевые) – пароли хранятся на веб-сайтах провайдеров;

− десктопные – пароли хранятся на локальном диске компьютера;

− портативные – пароли хранятся на мобильных устройствах или USB-накопителе.

1.1 Анализ существующих решений менеджеров паролей

В качестве готовых решений мобильных парольных менеджеров рассмотрим такие популярные приложения, как Dashlane, 1Password, RoboForm, Keeper, LastPass.

Dashlane использует алгоритм шифрования AES для сокрытия данных и двухфакторную аутентификацию, в которую входят и биометрические данные. Он обладает автоматическое сменой паролей после проверки всего

хранилища, т.е. изменяет слабые пароли на более надежные. Еще одной функцией является предоставление доступа пользователям к VPN.

По бесплатному тарифу Dashlane позволяет хранить около 50 паролей на одном устройстве. Также он имеет и платные тарифы Premium для одного или шести пользователей (семейный). В таком случае ежемесячная оплата составляет примерно 480 рублей.

1Password предоставляет такие функции, как:

− сканирование публичных баз данных на наличие утечек логинов и конфиденциальных данных;

− генерация надежных паролей;

− синхронизация с приложениями разовых паролей;

− использование биометрических данных и встроенный аутентификатор.

Данное решение не имеет бесплатного тарифа, только 14-дневный пробный период. Подписка в месяц составляет примерно 220 рублей.

RoboForm предоставляет возможность удобного заполнения любых электронных форм. Дополнением к этой функции является наличие нескольких «аккаунтов» для веб-форм. Помимо данных о пароле, формы автоматически заполняются паспортными данными, банковскими картами и тд.

Несколько дополнительных функций:

− сканер хранилища на наличие слабых и повторяющихся паролей;

− совместимость с приложениями для двухфакторной аутентификации;

− хранение и синхронизация закладок браузера на тех устройствах, где установлено данное приложение.

Бесплатный тариф предлагает заполнение форм, сканирование паролей на надежность, хранение закладок. За подписку около 130 рублей в месяц это дополняется синхронизацией неограниченного числа устройств, двухфакторной аутентификацией и облачным резервным копированием. Подписка для семьи за 2,400 рублей в год предоставляет все вышеперечисленные функции для пяти пользователей.

Keeper использует алгоритм шифрования AES, многофакторную аутентификацию, которая включает в себя биометрические данные. Предлагает сохранить пароли во время их ввода в какую-либо веб-форму, а также заполнение тех самых форм.

Данное решение включает в себя зашифрованные мессенджер с приватной галерей для сохранения файлов и ограниченным временем жизни сообщения.

Бесплатный тариф предоставляет ограниченное число функций и только одно устройство. Подписка за 3,400 рублей в год позволяет сохранять неограниченное количество паролей на неограниченном числе устройств, использовать многофакторную аутентификацию. Подписка за 7,400 рублей к год позволяет использовать вышеперечисленные функции пяти пользователям, увеличивает объем хранилища и мониторит публичные базы данных на предмет утечки данных.

LastPass в своем бесплатном тарифе позволяет хранить неограниченное число паролей на одном устройстве. Он автоматически сменяет ненадежные пароли на безопасные и предоставляет возможность ограниченной многофакторной аутентификации.

Тариф за 220 рублей в месяц позволяет использовать расширенную многофакторную аутентификацию, облачное хранилище, мониторинг публичных баз данных. Тариф за 290 рублей в месяц позволяет использовать все функции приложения шести пользователям лицензии.

1.2 Обоснование для разработки собственного программного продукта

Все вышеперечисленные системы являются хорошим выбором в своей категории. Но это не означает, что все способны воспользоваться данными предложениями. Данные приложения не предоставляют пользователю полный набор функций, обрекая пользователя на ежемесячные или ежегодные выплаты. Не каждый пользователь способен позволить себе оплату за хранения паролей для входа на сервисы.

Еще одним минусом является глобальный сервер, который без должной защиты станет легкой мишенью для злоумышленника. Как упоминалось ранее, при получении доступа к данным сервера, злоумышленник получит все персональные данные пользователей системы и доступ к их платежным системам. Такого допускать нельзя. Пользователь должен иметь возможность хранения паролей на локальном сервере, не отменяя также мер защиты. Таким образом, данные пользователя не смогут выйти дальше его локальной сети, что гарантирует большую безопасность, нежели глобальный сервер.

2 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

2.1 Цели и функции приложения

Целью разрабатываемого приложения в целом является хранение, обработка, сокрытие аутентификационных данных различных пользователей для необходимых ему сервисов.

Функции, которыми будет обладать готовое приложение:

− регистрация и авторизация пользователя;

− удаление аккаунта;

− сохранение, изменение и удаление аутентификационных данных;

− поиск по существующим данным;

− генерация случайных паролей с фильтрацией;

− обмен паролей между мобильным устройством и сервером, между сервером и клиентом на компьютере.

2.2 Требования к программе

Основные требования, помимо описанных ниже, представлены в Приложении А (Техническом задании на разработку программного обеспечения).

2.2.1 Общие требования

Разработанный продукт должен состоять из функциональных модулей, которые представлены в пункте 2.4, что обеспечит простое изменение и обновление модулей, добавление новых.

Программа должна осуществлять шифрование аутентификационных данных (логина и пароля) для баз данных на сервере и мобильном устройстве. Также необходимо обеспечивать защищенную передачу данных между клиентскими и серверным частями приложения.

Программа должна генерировать надежные пароли случайным образом на клиентском устройстве, не передавая данные в глобальную сеть.

Интерфейс приложения не должен вводить в заблуждение, иметь сложные двусмысленные фразы. Цветовая гамма не должна быть резкой, вызывающей раздражение. Весь текст должен быть виден и читаем.

2.2.2 Требования к лингвистическому обеспечению

Интерфейс программы состоит из полностью русифицированных фраз для обеспечения понимания смысла прочитанного.

2.2.3 Требования к математическому обеспечению

Программа включает в себя алгоритм расчета примерного времени подбора пароля (сгенерированного и самостоятельно набранного). Данный алгоритм определяет время на основании всех возможных паролей и позиции выбранного пароля из списка возможных. Формула для расчета представлена ниже:

$$T = \frac{M - N_{\text{пароль}}}{\upsilon} \quad (1)$$

где

$$M = A^{l} \quad (2)$$

Принятые обозначения:

T – время подбора пароля;

M – все возможные пароли;

$N_{\text{пароль}}$ – номер пароля в списке возможных паролей;

υ – скорость подбора пароля;

A – число символов в результирующем алфавите;

l – длина пароля.

2.2.4 Требования к техническому обеспечению

Мобильная часть приложения, при необходимости, способна автономно работать без серверной части. Тогда пользователь лишается возможности резервной копии данных на другом устройстве.

Минимальные требования для мобильной устройства:

− операционная система: Android 8 и выше, iOS 10 и выше;

− оперативная память: 1 Гигабайт;

− занимаемая память: 50 МБ Мегабайт.

Доступ к локальной сети при необходимости резервного копирования. Тогда минимальные требования для компьютера имеют вид:

− операционная система: Windows 10 и выше;

− оперативная память: 100 Мегабайт;

− занимаемая память: 80 Мегабайт.

Обязательное наличие локальной сети для подключения мобильного устройства.

2.3 Архитектура приложения

2.3.1 Архитектура приложения в целом

Для определения архитектуры программного продукта, необходимо рассмотреть несколько существующих шаблонов.

Многоуровневый шаблон используется для структурирования программ системы в зависимости от задач той или иной подсистемы. В настоящее время информационные системы данной архитектуры имеют четыре уровня абстракции:

− пользовательский интерфейс;

− уровень приложения (сервиса);

− уровень предметной области;

− уровень хранения данных.

Используется такая архитектура в веб-приложениях и на площадках электронной коммерции.

Клиент-серверный шаблон состоит из двух компонентов: сервер и клиент. От клиентов поступают запросы к серверу на выполнение тех или иных действий (работы с данными, предоставление данных и тд.), а тот отвечает необходимой информацией или действием. Сервер находится в постоянном прослушивании запросов.

Используются в приложениях, которым необходим доступ к интернету (электронная почта, интернет-банки и тд).

Одноранговый шаблон, в котором компоненты способны выступать в роли сервера и клиента в зависимости от необходимых действий и данных. Клиент может пользоваться «услугами» сервера, и, при необходимости, выступать в качестве сервера для других клиентов.

Такая архитектура используется в файлообменных сетях и мультимедийных протоколах.

В результате анализа была выбрана клиент-серверная архитектура приложения, так как разрабатываемая система имеет серверную и клиентскую части, которые будут взаимодействовать по локальной сети. Роли каждой из этих частей строго определены.

Для данного проекта схема клиент-серверной архитектуры будет иметь вид, представленный на рисунке 2.1.

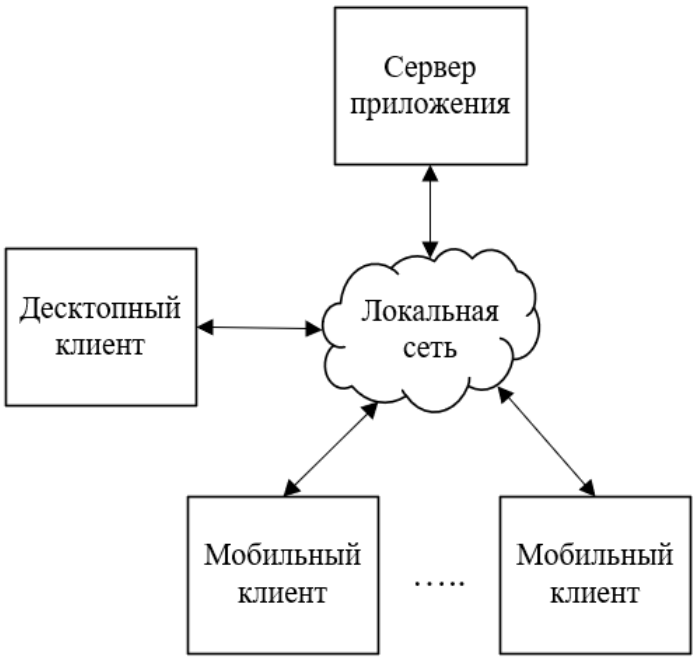

Рисунок 2.1 – Схема клиент-серверной архитектуры

Схема архитектуры включает в себя только одного клиента для компьютера, потому что данная часть используется для регистрации пользователя и просмотра данных. Может быть использована несколькими пользователями, то есть обладает многопользовательским режимом. Детальное описание работы данной схемы представлено в пункте 2.4.

2.3.2 Архитектура каждой подпрограммы в отдельности

В качестве шаблона для разработки клиентских приложений для мобильного устройства и компьютера был взят шаблона MVVM.

MVVM – шаблон, в котором основными компонентами являются модель (содержит бизнес-логику приложения), представление (содержит пользовательский интерфейс) и представление-модель (содержит логику пользовательского интерфейса). На рисунке 2.2 показана схема взаимодействия компонентов архитектурного шаблона.

На схеме показано также отношения модели-представления с командами и поведениями. Так как View-Model представляет собой логику для View, компонент содержит описание команд и поведений, которые будут происходить в представлении.

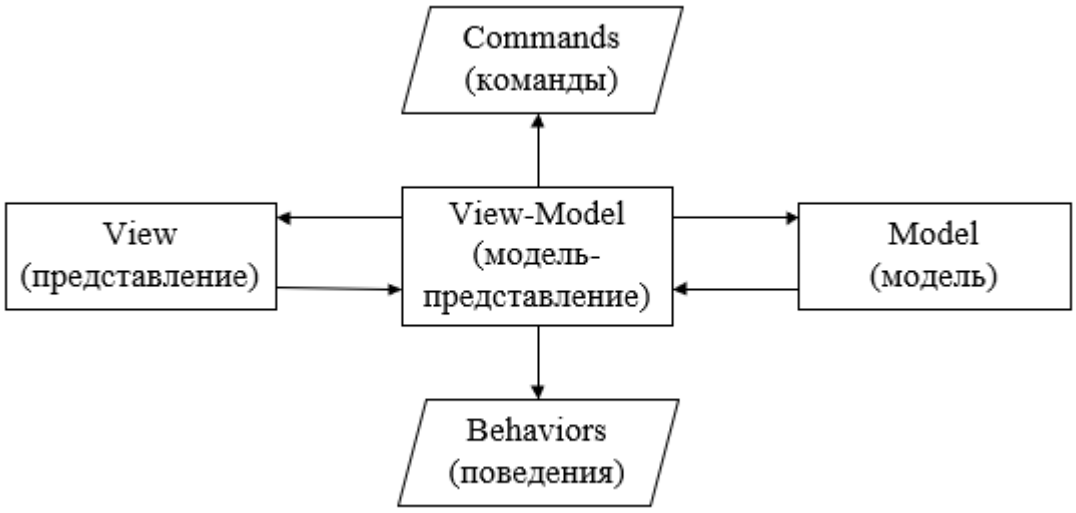

Рисунок 2.2 – Архитектурный шаблон MVVM

Главной целью разделения вышеупомянутых компонентов заключается в ускорении разработки и сопровождения. Объясняется это тем, что при изменении одного из компонентов, не затрагивается другой.

2.4 Структура приложения

Схема работы приложений как единой системы представлена на рисунке 2.3. Данная схема содержит мобильное, серверное и десктопное приложение. Взаимодействие этих частей происходит за счет HTTP запросов (HTTPGET, HTTPPOST, HTTPPATCH, HTTPDELETE). На все приложения накладывается валидатор, не позволяющие сохранить некорректные данные. Серверная часть

выступает в качестве веб-сервера, поэтому он управляется таким параметром, как Веб API.

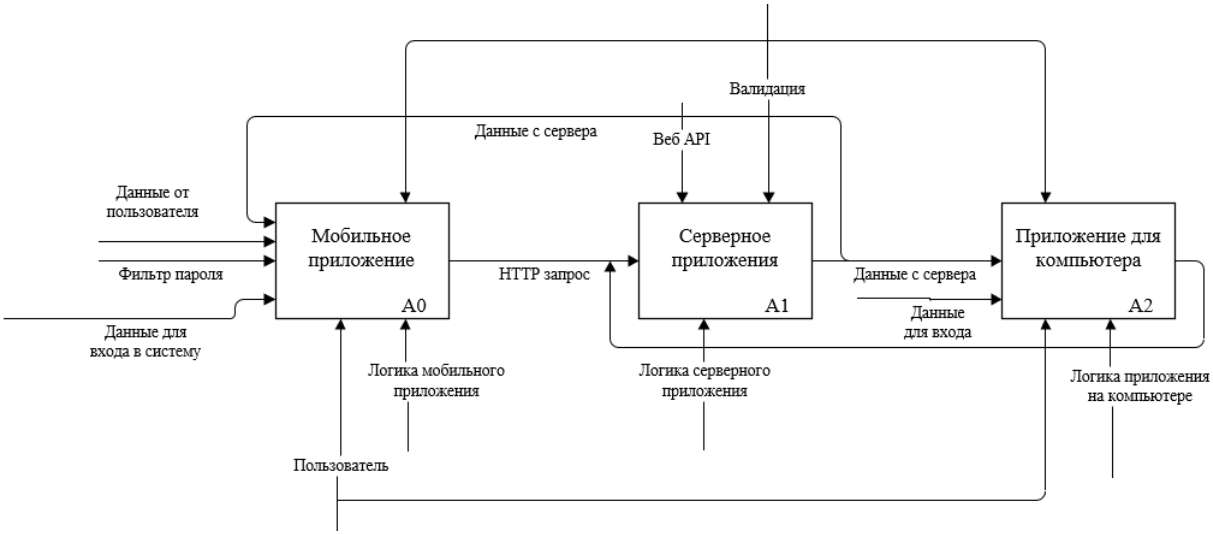

Рисунок 2.3 – Схема работы приложения как системы

Пользователь участвует в работе двух клиентских приложений, то есть в мобильном и компьютерном. Данными от пользователя являются хранимые аутентификационные данные и данные для входа в систему.

Детально рассмотрим работу каждого их приложений. На рисунке 2.4 представлена схема работы мобильного приложения.

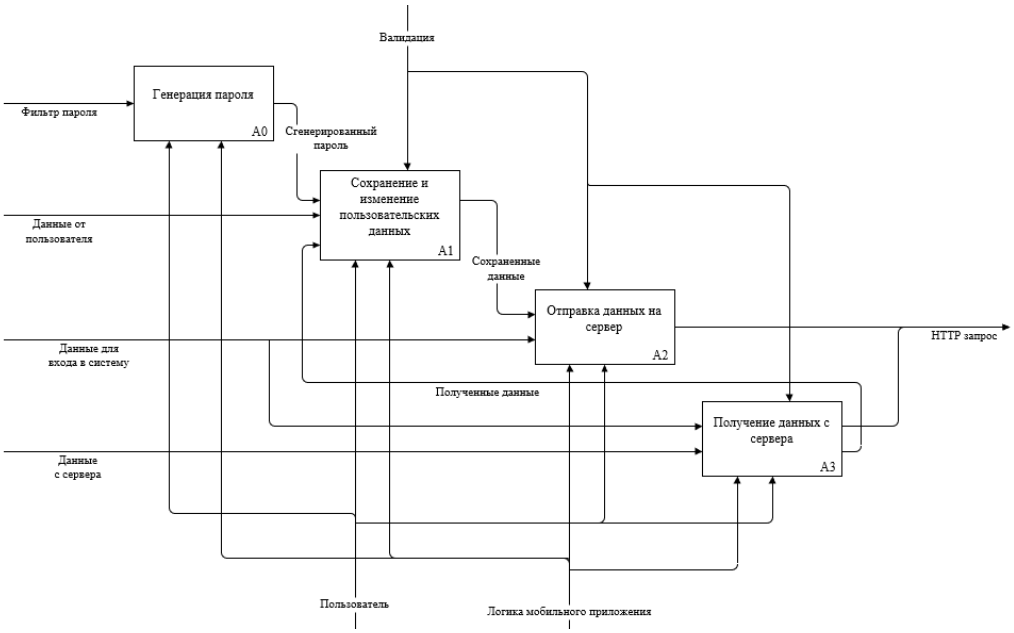

Рисунок 2.4 – Схема работы мобильного приложения

Работа мобильного приложения заключается в создании и хранении пользовательских данных, инструментом которых является генерация случайных паролей. Помимо этого, пользователь может оправлять данные на сервер в качестве резервных копий и получать на мобильное устройство после ввода логина и пароля для входа в систему.

На рисунке 2.5 представлена схема работы приложения для компьютера.

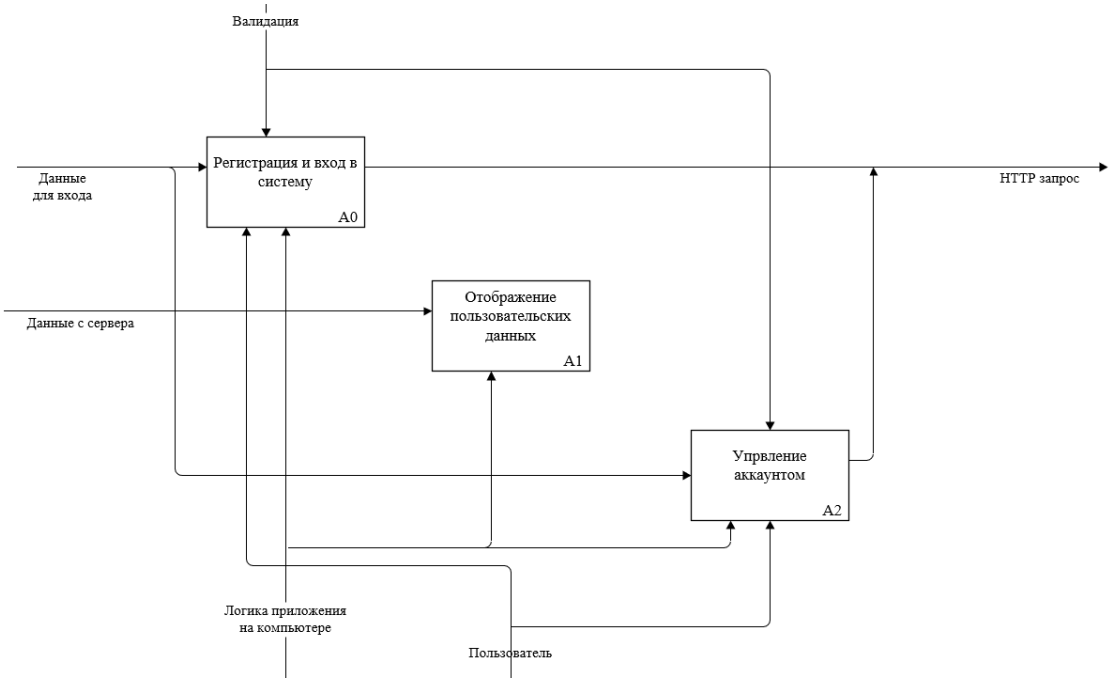

Рисунок 2.5 – Схема работы приложения на компьютере

Основная работа компьютерного приложения заключается в регистрации пользователя и предоставления данных после входа в систему. Оно полностью зависит от серверной части, так как для работы ему необходимо получать данные о зарегистрированных пользователях, запрашивать их данные, управлять аккаунтом.

На рисунке 2.6 представлена схема работы серверного приложения. Основана работа на веб-контроллера, которые находятся в режим прослушивания запросов. Контроллер аккаунта занимается регистрацией, аутентификацией и авторизацией, регистрацией мобильного устройства и удалением аккаунта. Контроллер паролей – получением и отправкой пользовательских данных.

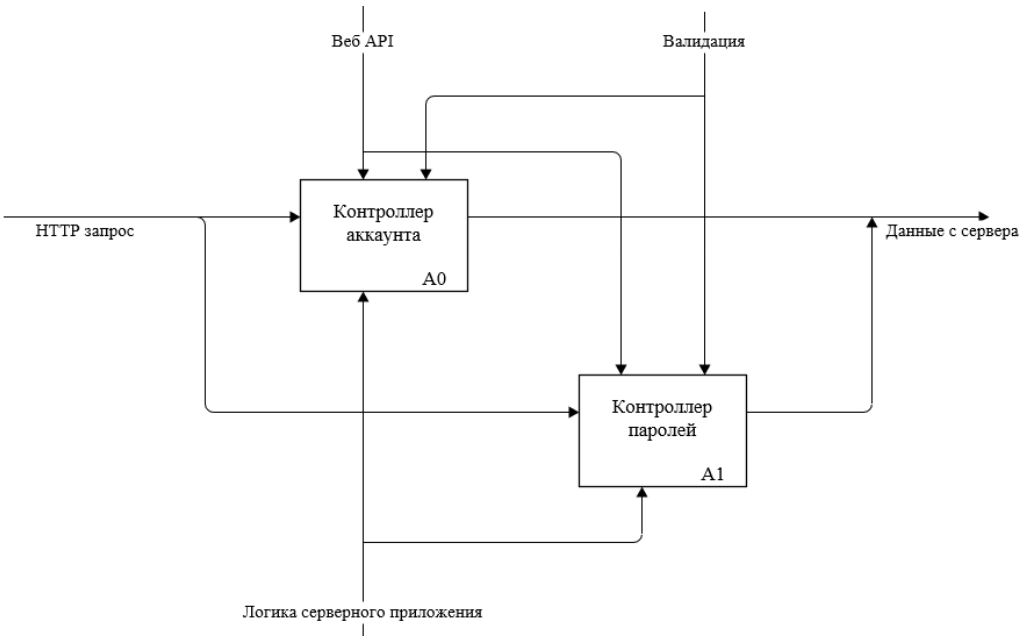

Рисунок 2.6 – Схема работы серверного приложения

Серверное приложение запускается параллельно с компьютерным, тем самым предоставляя ему данные о пользователях системы сразу при старте.

Детальное рассмотрение работы приложений рассмотрено в пункте 2.5.

2.5 Характеристика функциональных модулей программы

Приложение должно включать в себя следующие функциональные модули:

Модуль управления аккаунтом. Регистрация аккаунта и его удаление происходит на компьютерной части приложения. Мобильная часть оснащена только входом в систему для дальнейшей отправки данных на серверное приложение. Также мобильное приложение отправляет серверу сведения об устройстве, на котором произошел вход. Таким образом, пользователь после авторизации в компьютере может проверить мобильное устройство.

Данный модуль представляет собой контроллер с логикой в серверном приложении, где все операции с аккаунтом представлены в качестве HTTP запросов (добавление, изменение, получение, удаление). Схема управления аккаунтом представлена на рисунке 2.7.

Пользователь отправляет данные с клиентского приложения на серверное. Далее данные проверяются на корректность (удовлетворяют требованиям длины и алфавита) и сохраняются в базе данных, перед этим переводятся в

зашифрованный вид. Далее данные, обработанные сервером, отправляются пользователю. Если проверка на корректность не прошла успешно, сервер отправляет статус ошибки (NotFound, BadResult).

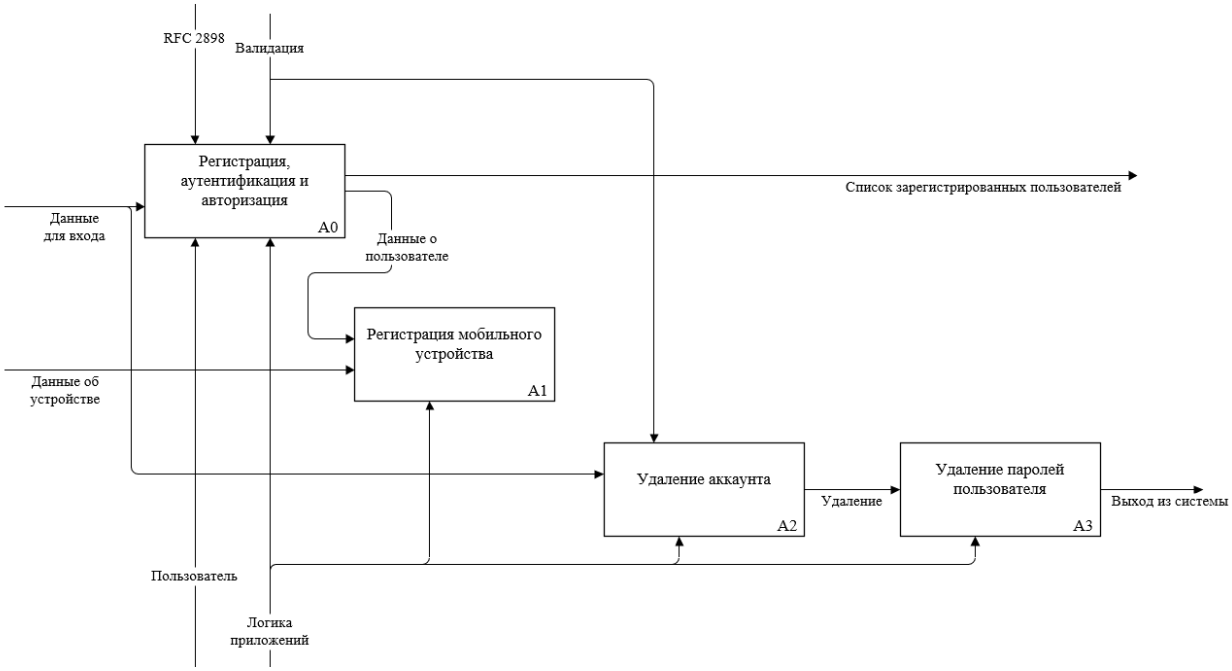

Рисунок 2.7 – Управление аккаунтом

При удалении аккаунта, после верного ввода логина и пароля, удаляются пользовательские данные с сервера.

Модуль управления паролями. Добавление и изменение паролей в мобильном устройстве, их удаление из базы данных. Также в данный модуль входит отправка на сервер и получение всех данных с сервера. Взаимодействие с сервером происходит через HTTP запросы (добавление и получение). Схема управления паролями показана на рисунке 2.8.

Перед сохранением в базу данных сервер и мобильное приложение шифруют данные, используя случайно сгенерированный ключ в первый запуск приложения. Для отображения данных пользователю или отправкой их с сервера информация расшифровывается.

Так как мобильное приложение может работать в автономном режиме, данные отправляются и получаются с сервера по запросу пользователя, а не при каждом добавлении данных. Данное решение обосновывается тем, что

добавление, изменение и удаление данных может происходить без подключения к серверному приложению.

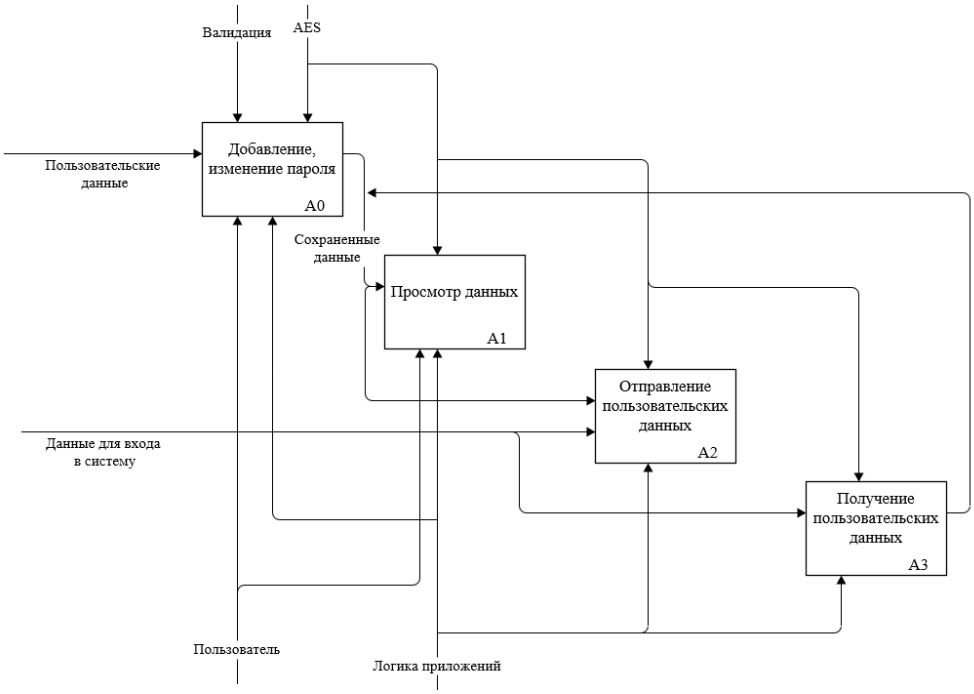

Рисунок 2.8 – Управление паролями

Перед отправкой и получением пользовательских данных запрашиваются данные для входа в систему, чтобы однозначно определить пользователя, которому эти данные принадлежат.

Модуль генерации пароля. Пароль генерируется на мобильном устройстве после выбора пользователем определенных фильтров (алфавит пароля и его длина). После происходит слияние алфавитных наборов для случайного выбора из полученного диапазона символов.

Помимо вышесказанного, данный модуль высчитывает примерное время подбора пароля методом грубой силы. Для этого определяется, из каких алфавитных наборов состоит пароль. Формула для вычисления представлена в пункте 2.2.3 (требования к математическому обеспечению).

Модуль клиент-серверного взаимодействия. Сервер выступает в качестве веб-сервера, из-за чего обращение происходит через протокол HTTPS, предусматривающий шифрование данных.

Мобильное приложение предусмотрено для работы вне сети, из-за чего, при отсутствии соединения, оно работает автономно. На рисунке 2.9 показана блок-схема взаимодействия мобильного клиента с сервером.

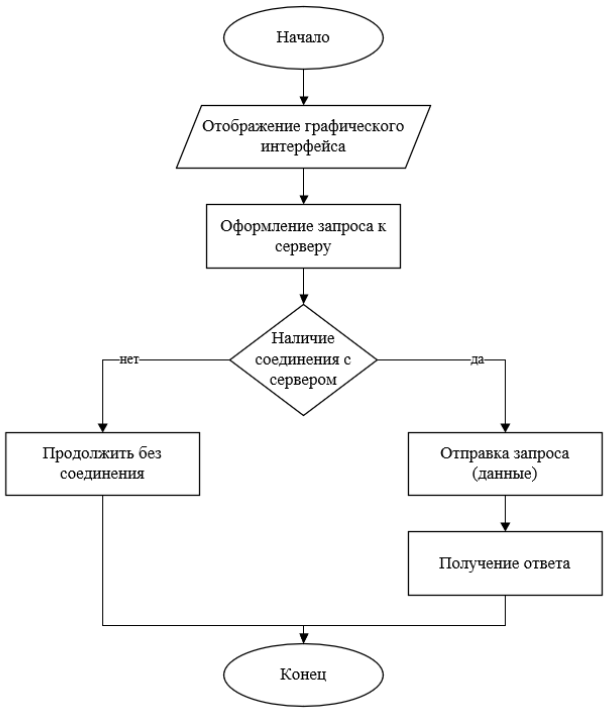

Рисунок 2.9 – Блок-схема клиент-серверного взаимодействия

То есть мобильное приложение отображает пользователю сообщение об ошибке подключения, не прекращая работу.

Клиент на компьютере предусмотрен только для просмотра данных с сервера, из-за чего ему необходимо постоянное соединение. Для этого возможно обращение на localhost, так как для этого используется только одна машина.

2.6 Проектирование базы данных

Для корректной работы приложения, данные необходимо хранить в базе данных на обоих устройствах системы. Из чего следует необходимость проектирования двух баз данных.

Для начала рассмотрим базу данных мобильного приложения. Она включает в себя одну таблицу «Пароль», в которой хранится название сервиса

(к которому применяется соответствующий пароль), логин, пароль и дата создания. В таблице 2.1 представлено детальное описание данной сущности.

Таблица 2.1 – Описание атрибутов таблицы «Пароль»

Наименование атрибута	Тип данных	Условия	Формат данных	Индексация
ID	Число	> 0	Integer	Primary key
Сервис	Текст	–	Varchar(max)	–
Логин	Текст	–	Varchar(max)	–
Пароль	Текст	–	Varchar(max)	–
Дата	Дата/Время	–	Varchar(max)	–

Далее рассмотрим модель данных. На рисунке 2.10 и 2.11 представлены логическая и физическая модели данных соответственно для мобильной базы данных (нотация IDEF1X).

Рисунок 2.10 – Логическая модель данных мобильного приложения

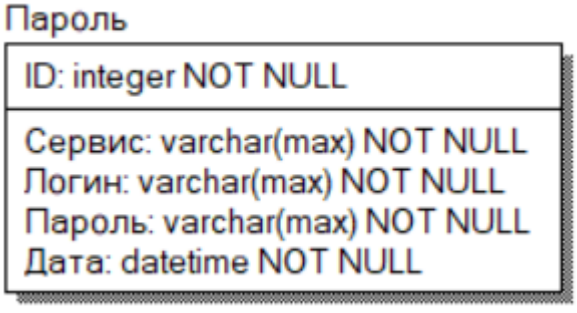

Рисунок 2.11 – Физическая модель данных мобильного приложения

В таблице «Пароль» предусмотрено поле с датой создания пароля, чтобы позже пользователь мог отследить, какие пароли являются устаревшими.

База данных состоит из одной таблицы, которая работает только с несвязанными данными. В дальнейшем возможна реализация пользовательских папок или категорий для структурированной огранизации аутентификационных данных.

Если пароль и логин будут храниться только в одной категории, тогда необходимо будет добавить только одну таблицу, организовав между ними связь «один-ко-многим». Если же они будут располагаться в нескольких категориях, тогда таблиц на добавление будет две, для организации связи «многие-ко-многим».

Далее рассмотрим базу данных серверной части данного приложения. Она включает в себя две таблицы: «Пароль» и «Аккаунт». Первая сущность схожа с сущностью на мобильной версии, за исключением наличия в ней внешнего ключа от сущности «Аккаунт». Вторая – данные о пользователе: его логин, пароли и название мобильного устройства, с которого произошел вход. В таблице 2.2 и 2.3 детально описаны сущности «Аккаунт» и «Пароль» соответственно.

Таблица 2.2 – Описание атрибутов таблицы «Аккаунт»

Наименование атрибута	Тип данных	Условия	Формат данных	Индексация
ID	Число	> 0	Integer	Primary key
Логин	Текст	–	Varchar(max)	–
Пароль	Текст	–	Varchar(max)	–
Устройство	Текст	–	Varchar(max)	–

Таблица 2.3 – Описание атрибутов таблицы «Пароль»

Наименование атрибута	Тип данных	Условия	Формат данных	Индексация
ID	Число	> 0	Integer	Primary key
Сервис	Текст	–	Varchar(max)	–
Логин	Текст	–	Varchar(max)	–
Пароль	Текст	–	Varchar(max)	–
ID Аккаунта	Число	–	Integer	Foreign key

Логическая и физическая модели данных представлены на рисунке 2.12 и 2.13 соответственно (нотация IDEF1X). Сущности «Аккаунт» и «Пароль» имеют связь «один-ко-многим».

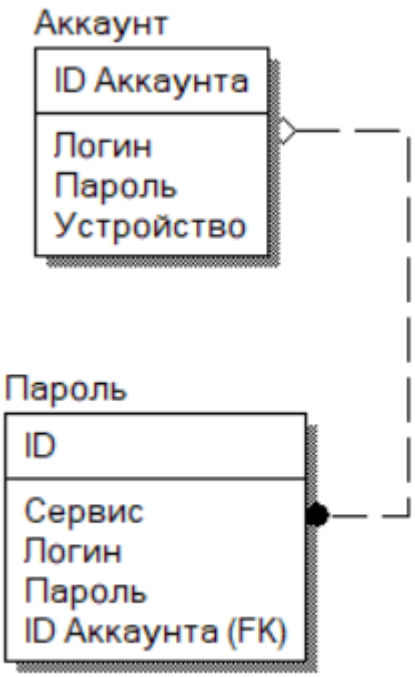

Рисунок 2.12 – Логическая модель данных серверного приложения

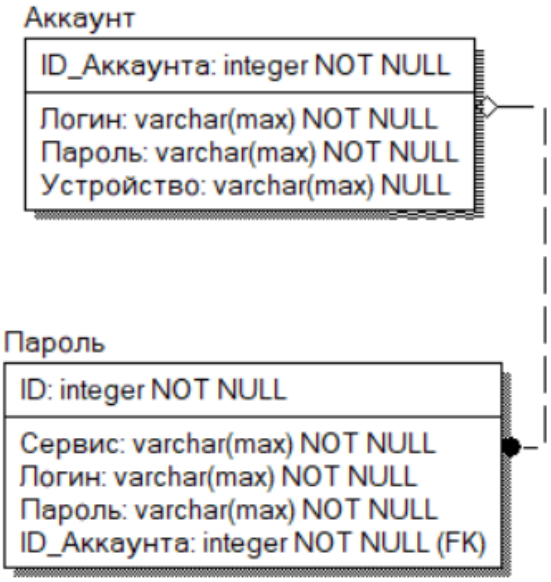

Рисунок 2.13 – Физическая модель данных серверного приложения

Поле «Устройство» может иметь значение NULL, так как при создании аккаунта пользователь еще не регистрирует мобильное устройство.

Так же возможно добавление пользовательских категорий, описанных выше. Тогда база данных возрастет на одну или две таблицы. Но при этом необходимо будет определять, какие категории были созданы тем или иным пользователем.

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

3.1 Методология разработки программного продукта

В качестве методологии разработки была выбрана «V-Model», или модель «шаг за шагом», которая была унаследована от каскадной модели. Данная методология применима в системах, где важна непрерывная работа. Схема модели представлена на рисунке 3.1.

Направлена эта модель на постоянные проверки и тестирования продукта на всех этапах программирования, то есть параллельно с кодированием. Помимо этого, требования к системе должны быть строго определены в начале работы.

«V-Model» схожа с моделью экстремального проектирования. После добавления какого-либо объекта или подсистемы (детализации проекта) происходит тестирование. Далее результаты тестирования утверждаются, и продолжается процесс кодирования системы.

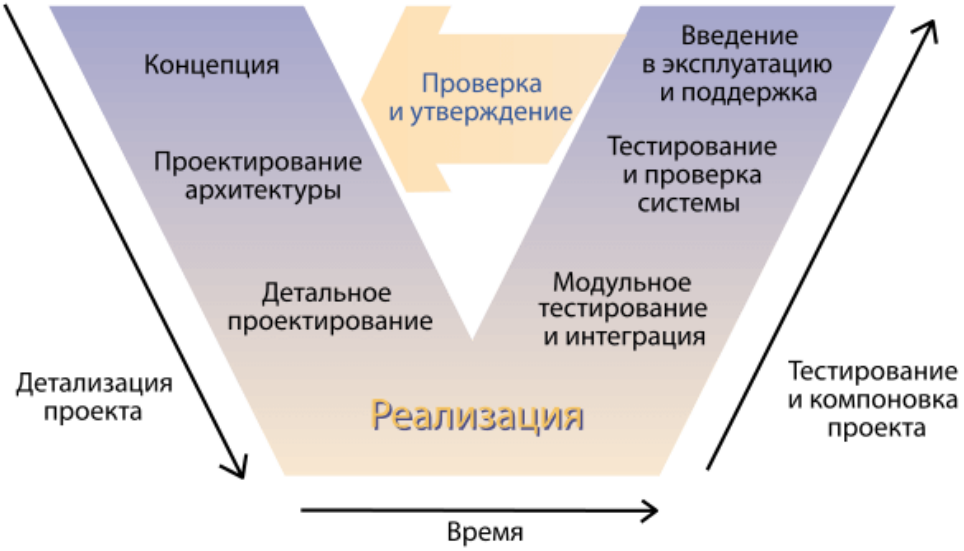

Рисунок 3.1 – Модель «V-Model»

Разработка и тестирование системы начиналось с мобильного приложения. Добавлялись новые экранные формы и логика к ним, новые программные

модули. После добавления каждого нового элемента приложение подвергалось тестированию.

Далее разрабатывалось и тестировалось серверное приложение, сетевое взаимодействие между ним и мобильным приложением. Добавлялись методы и контроллеры, которые после тестирования утверждались, после чего разработка переходила к следующим компонентам.

Последним разрабатывалось клиентское приложение для компьютера. Когда вся система была готова, тестировалась ее общая работа, исправлялись незначительные ошибки.

3.2 Средства разработки

Для корректной работы приложения, необходимо три программы: мобильный клиент, с которым пользователь работает большую часть времени; серверная часть, на которую передаются данные на хранение; десктопный клиент, который просматривает пароли пользователя с сервера. Рассмотрим языки программирования и возможные технологии разработки, чтобы определить подходящий.

Первым рассмотрим мобильную разработку. Для анализа были взяты три популярные технологии разработки кроссплатформенных мобильных приложений. В таблице 3.1 представлены основные характеристики этих технологий и необходимых языков программирования.

Таблица 3.1 – Технологии мобильной разработки

Название технологии	Языки разработки	Компания	Плюсы	Минусы
React Native	Java Script React.js	Facebook	Большое количество встроенных библиотек	Сложность разработки
Flutter	Dart *Поддерживает* Swift, C, Java и тд	Google	Быстрая и простая разработка	Не обеспечивается поддержка готовых библиотек
Xamarin	C#	Microsoft	Сниженные стоимость и скорость разработки за счет общего проекта	Необходимы знания Java, Objective-C, Swift и тд.

Оценив представленные выше технологии, выбрана технология Xamarin, использующая язык C# и фреймворк .Net Standard 2.1. Для разработки одного программы создается несколько проектов: общий проект, в котором пишется логика приложения и добавляются экранные формы; проекты под каждую из платформ, в которых добавляется код для работы со специфичными для каждой операционной системы функциями.

Данная технология способна разрабатывать программы для Android, iOS, Windows совместимые.

Язык программирования для общего приложения был выбран объектно-ориентированный C#, который подходит не только для мобильной разработки, но и для веб-приложений, веб-серверов, десктопных программ. Отсюда следует, что для серверной и десктопной части менеджера паролей были выбраны технологии, использующие язык C#.

Для десктопной части использовался WPF, использующий фреймворк .Net Core 3.1. Данная технология используется для разработки Windows приложений.

Помимо C#, технологии Xamarin и WPF используют язык разметки XAML, который предназначен для отрисовки графического интерфейса.

Для серверной части использовался ASP .NET Core, использующий фреймворк .Net 6. Данная технология представляет из себя веб-разработку, но в рамках менеджера паролей использовались только контроллеры без экранных форм.

Так как в программе предусматривается хранение конфиденциальных данных, необходимо использовать систему управления базами данных. Для такой задачи целесообразно выбирать реляционные базы данных; в противным же случае данные будут передаваться со значительным снижением скорости.

Таким образом, в качестве СУБД была взяла SQLite. Несмотря на небольшое количество реализованных типов данных (integer, real, text, blob), она способна работать с необходимыми задачами.

В качестве среды разработки, основываясь на вышеупомянутых языке и технологиях, была выбрана Visual Studio 2022 Community.

3.3 Структура разработанного проекта

Вся система состоит из трех основных проектов: мобильное приложение, серверное приложение и приложение для компьютера; и двух дополнительных проектов: для ОС Android, для ОС iOS.

Проекты для мобильных операционных систем представлены на рисунке 3.2 Они состоят из пользовательских отрисовщиков (EntryRenderer), переопределения всплывающих сообщений и дополнительных стартовых файлов для работы.

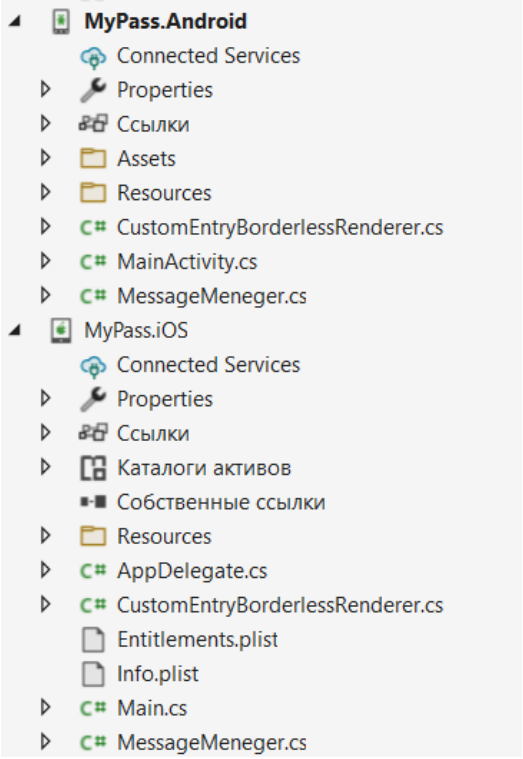

Рисунок 3.2 – Проекты для Android и iOS

Без необходимости данные проекты не подлежат изменению.

Состав основного проекта мобильного приложения представлен на рисунке 3.3. В нем пишется вся логика приложения для телефона, добавляются экранные формы.

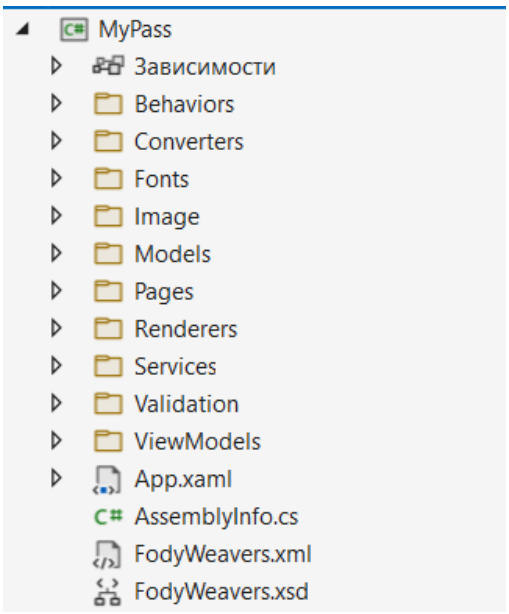

Рисунок 3.3 – Основной проект мобильного приложения

Состоит данный проект их поведений, конверторов, визуализаторов и валидатора (рисунок 3.4); шрифтов и иллюстраций; дополнительных файлов для работы, таких как обращение к серверу, к базе данных, шифрование и тд, расположенных в каталоге Services.

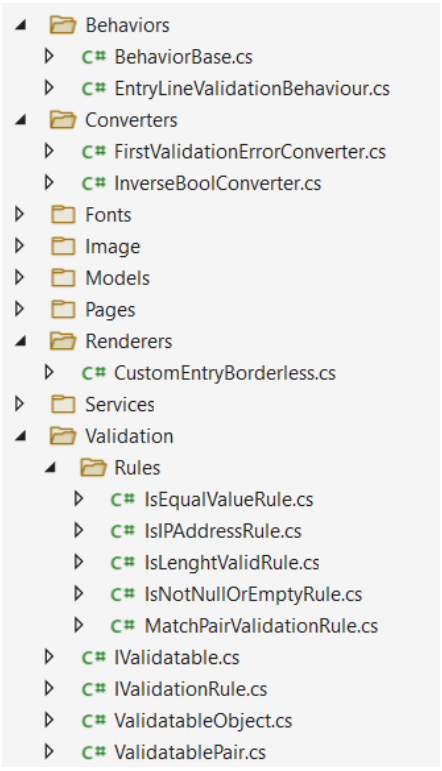

Рисунок 3.4 – Файлы поведения, конвертеров, визуализаторов и валидатора мобильного приложения

Основные файлы проекта расположены в каталогах Model, ViewModel и Page, организуя шаблон MVVM, описанный в пункте 2.3.2. Файлы экранных форм и логики этих форм представлены на рисунке 3.5.

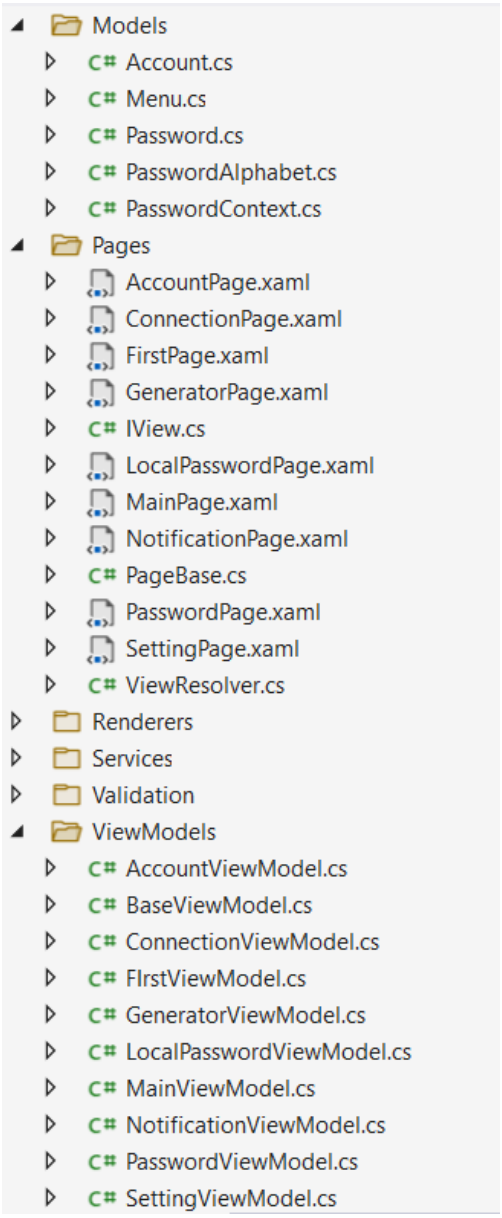

Рисунок 3.5 – Файлы экранных форм и логики

Помимо экранных форм и файл логики, данные каталоги содержат дополнительные файлы, необходимые для шаблона MVVM, разделяющие данные компоненты. Они позволяют обращаться к таким командам страниц, как закрытие и открытие модальной и обычной страницы.

Далее рассмотрим проекты серверного приложения и приложения для компьютера. На рисунке 3.6 показаны важные файлы и каталоги серверной части. На рисунке 3.7 показаны файлы десктопной части.

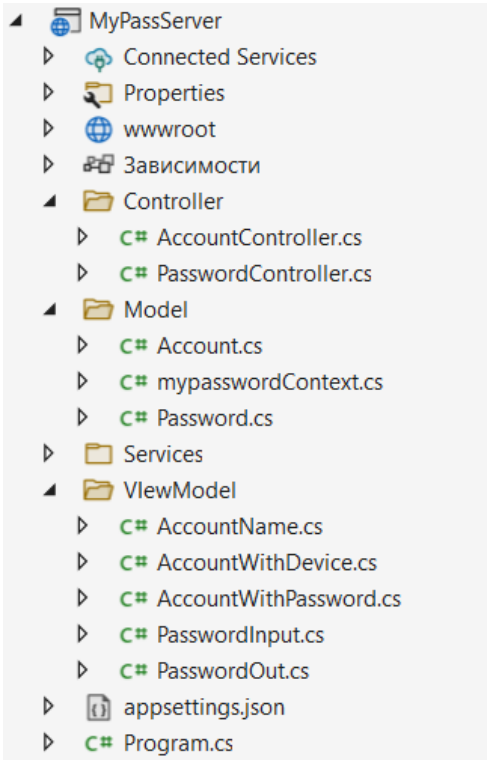

Рисунок 3.6 – Файлы и каталоги серверного приложения

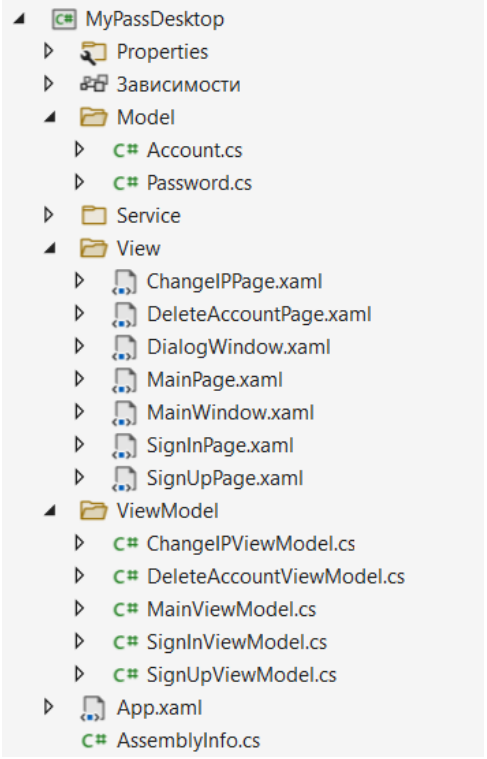

Рисунок 3.7 – Файлы и каталоги десктопного приложения

В состав серверного проекта входят такие файлы, как контроллеры, отвечающие на HTTP запросы; файлы обращения к базе данных, ее открытие и определение ключей; и дополнительные файлы, необходимые для работы.

Данное приложение не имеет экранных форм из-за ненадобности. Иначе данный проект имел бы шаблон MVC (Model-View-Controller).

В состав десктопного проекта входят окна и страницы, логика для страниц и другие дополнительные файлы, необходимые для работы приложения. Приложение строится на шаблоне MVVM, поэтому в проект также включены каталоги Model, View и ViewModel.

3.4 Описание работы приложения

При открытии приложения в первый раз, пользователь может наблюдать приветственную станицу, представленную на рисунке 3.8. Данная страница не несет никакой смысловой нагрузки на пользователя. Наличие данной страницы является хорошим тоном.

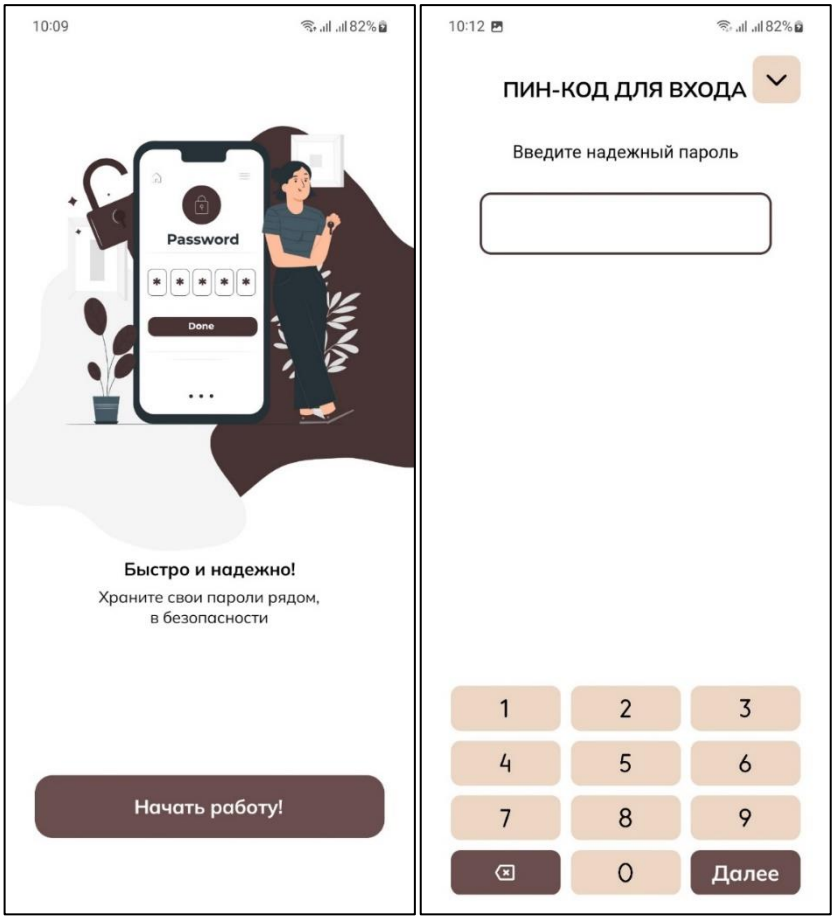

Рисунок 3.8 – Приветственная страница и страница входа с паролем

Также на рисунке 3.8 представлена страница входа в приложение, когда пользователь добавляет пароль для входа. Данная страница появляется сразу после запуска и открывает пользователю доступ к данным только после ввода пароля.

Далее пользователь перенаправляется на главную страницу приложения (рисунок 3.9), в котором он, после добавления паролей, будет просматривать свои аутентификационные данные. Также на этой странице присутствует явное меню навигации по приложению, что позволяет пользователю перейти на страницу генератора паролей и настроек. Помимо этого, пользователь имеет возможность перейти на страницы уведомления и создания пароля.

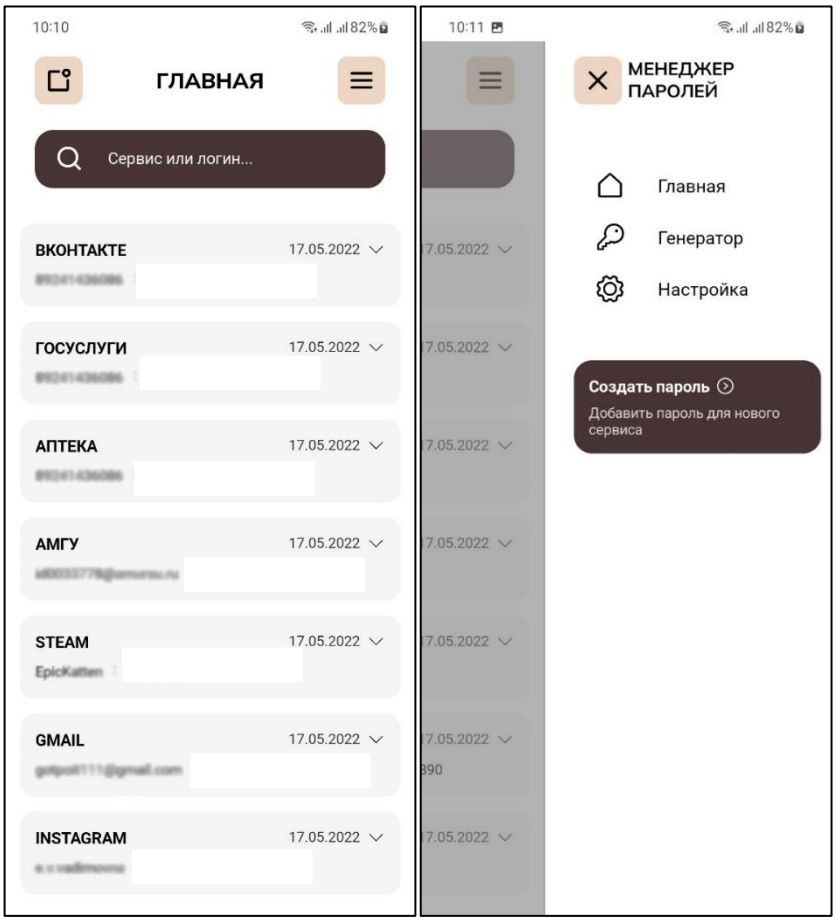

Рисунок 3.9 – Главная страница приложения с меню навигации

Через меню навигации возможно перейти на страницу генератора пароля, представленную на рисунке 3.10. Данная страница включает в себя фильтрацию алфавита и ползунок для выбора длины пароля. После выбора всех

необходимых фильтров, генерируется пароль и отображается примерное время его взлома по формуле, представленной в пункте 2.2.3. Далее пользователь может создать запись с данным паролем и скопировать его для использования.

Рисунок 3.10 – Генератор паролей

К фильтрам относятся такие параметры, как наличие строчных и заглавных букв, цифр и специальных символов. Последний параметр – длина, которая меняется при изменении положения ползунка.

Для отправления данных на компьютер, необходимо создать аккаунт через клиентское приложение на компьютере. Страница регистрации представлено на рисунке 3.11. Для создания аккаунта достаточно ввести логин и дважды пароль.

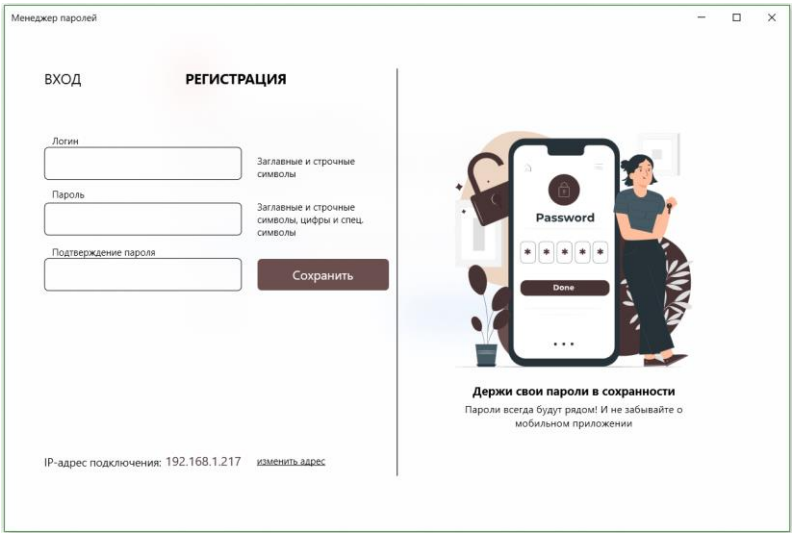

Рисунок 3.11 – Страница регистрации аккаунта

Далее пользователь переходит на страницу управления аккаунтом и просмотра паролей, представленную на рисунке 3.12. Как и в уведомлениях на мобильном устройстве, на данной странице отображаются устаревшие пароли (пароли, срок действия которых превышает полгода).

После того, как аккаунт создан, пользователь должен войти в него через мобильное устройство. Для этого необходимо перейти в «Настройки» через меню навигации и выбрать «Войти в аккаунт». Далее пользователь вводит логин и пароль. В случае ошибки появляется окно с ее описанием и примерной причиной. Страница настроек представлена на рисунке 3.13.

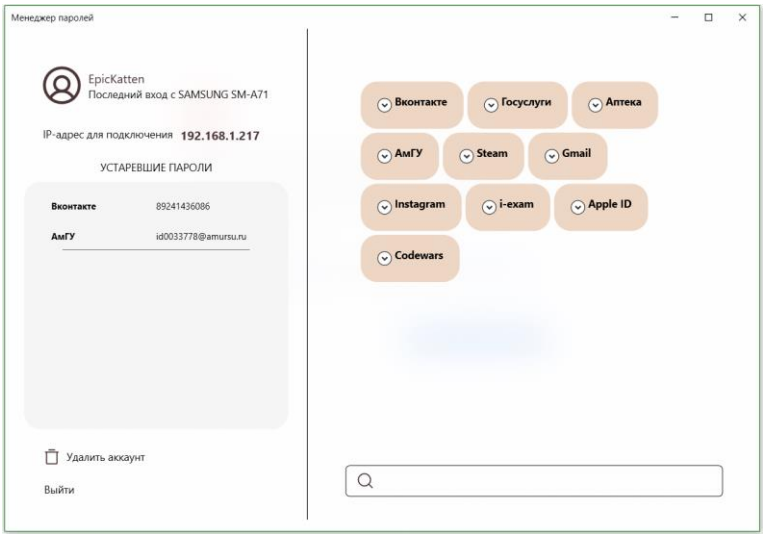

Рисунок 3.12 – Главная страница приложения

Перед тем, как войти в аккаунт, пользователю необходимо ввести IP-адрес компьютера, на котором находится сервер. Для этого на странице «Настройки» выбирается пункт «Изменить адрес подключения». Страница изменения IP-адреса также представлена на рисунке 3.13.

После входа в аккаунт на мобильном устройстве, на сервер отправляется модель и название устройства для дальнейшего отслеживания пользователем. На главное странице приложения на компьютере отображается устройство, с которого происходил последний вход в систему.

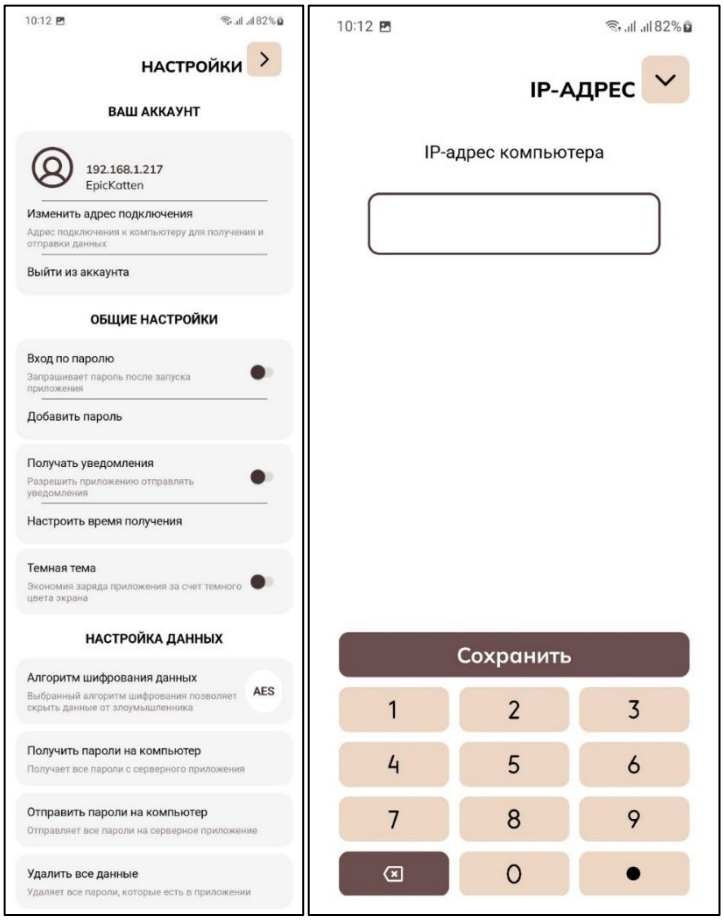

Рисунок 3.13 – Страница настроек и изменения IP-адреса

Помимо вышеперечисленного, страница настроек обладает такими функциями, как удаление всех данных с мобильного устройства.

3.5 Пути развития программного продукта

В качестве дальнейшей возможной модификации можно рассмотреть добавление таких функций в программу, как:

Автозаполнение электронных полей ввода. Данная функция упрощает работу пользователя с сервисами, позволяя заполнить поля, что, в свою очередь, ускоряет процесс авторизации, регистрации, идентификации и аутентификации.

Помимо заполнения полей с аутентификационными данными, данная функция может заполнять поля с паспортными данными и другими документами, которые пользователь введет в систему. Это также ускорит заполнение важных электронных документов.

Защита от фишинга за счет встроенной аутентификации. Функция позволит пользователю не вводить данные при входе в систему, так как приложение само будет связываться с системами аутентификации. Данная функция не только упросит пользователю вход в системы, но и защитит от фишинга, так как система не позволит авторизоваться пользователю на фальшивом сервисе.

Переопределение и внедрение системы в организации. При определении системы для организации основной ее функцией станет выдача паролей сотрудникам. Те, в свою очередь, при авторизации в приложении для компьютера, будут видеть все пароли, которые необходимы им для работы в системах организации.

Двухфакторная аутентификация при внедрении системы в организацию. Двухфакторная аутентификация – метод аутентификации в системе, при которой пользователь предоставляет два набора аутентификационных данных: логин/пароль и электронная почта/SMS. Данный метод позволит пользователю использовать одни из вышеперечисленных аутентификационных данных при входе в систему для просмотра паролей от программ организации.

4 ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

4.1 Реализация информационной защиты разработанного приложения

В состав приложения входят две базы данных (на мобильном устройстве и компьютере), в которых происходит шифрование аутентификационных данных, то есть паролей и логинов. Для этого используется алгоритм блочного шифрования AES с 256-битным ключом. В качестве ключа на мобильном устройстве используется случайный массив байт, который генерируется при первом запуске программы. В серверном приложении в качестве ключа выступает пароль пользователя, преобразованный в 256-битный массив. Код шифрование и дешифрование представлены на рисунках В.1 и В.2 соответственно, которые представлены в Приложении В.

Пароль пользователя, рассмотренный выше, выступает в качестве мастер-пароля, и хранится в базе данных в виде хэша, рассчитанного по стандарту RFC 2898 с применением соли. Данный стандарт представляет собой преобразование пароля в ключ шифрования. В целях безопасности мастер-пароль является неизменным, поэтому следует запомнить его. Код хэширования и проверки введенного пароля представлены на рисунках Г.1 и Г.2 соответственно, которые представлены в Приложении Г.

Серверное приложение выступает в качестве веб-сервера, поэтому обращения к нему происходят за счет HTTPS запросов, который является расширением протокола HTTP для поддержки шифрования. Следовательно, данные шифруются при передаче между частями системы.

Также не стоит забыть о том, что данные внутри системы передаются по локальной сети, не используя при этом выход в глобальную сеть.

4.2 Модель угроз информационной безопасности

Согласно «Методике оценки угроз безопасности информации» ФСТЭК, угроза безопасности информации есть совокупность условий и факторов, создающих опасность нарушения безопасности информации.

Обращаясь с стандарту ГОСТ Р 51275-2006, можно привести перечень факторов, способных воздействовать на защищаемую информацию приложения «Менеджер паролей».

Внутренние факторы включают в себя:

− разглашение информации имеющими доступ лицами, к которому относятся: передача информации по открытым линиям связи пользователями системы; передача носителя информации посторонним лицам; утрата носителя информации;

− неправомерные действия имеющих доступ лиц, к которым относятся: несанкционированное изменение или удаление данных других пользователей системы; несанкционированное копирование информации других пользователей системы;

− ошибки пользователей при эксплуатации системы, к которым относятся: удаление программы; удаление данных пользователей.

Внешние факторы включают в себя:

− доступ к информации с помощью технических средств;

− несанкционированный доступ к информации, к которому относится: использование закладных устройств; использование вредоносного программного обеспечения;

− блокирование доступа к информации с помощью перегрузки технических средств.

4.3 Модель нарушителя информационной безопасности

Нарушитель информационной безопасности – физическое лицо, случайно или преднамеренно совершившее действия, следствием которых является нарушение безопасности информации при ее обработке техническими средствами в информационных системах. В таблице 4.1 приведена модель возможных нарушителей информационной безопасности менеджера паролей.

Таблица 4.1 – Модель нарушителя

Нарушитель	Тип нарушителя	Квалификация нарушителя	Преследуемая цель	Возможности нарушителя
Пользователи системы	Внутренний	Низкая	Определить конфиденциальную информацию других пользователей	Случайный подбор пароля к учетной записи другого пользователя
		Средняя	Определить конфиденциальную информацию других пользователей, ее рассекречивание	Подбор пароля к учетной записи другого пользователя системы. Получение доступа к информации из базы данных при помощи специального ПО
Третьи лица, имеющие доступ к компьютеру пользователей	Внешний	Средний	Определить конфиденциальную информацию других пользователей, ее рассекречивание. Получение доступа к персональным данным за счет аутентификационных данных	Получение доступа к информации из базы данных при помощи специального ПО. Перехват HTTP запросов к системе во время работы

5 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

5.1 Безопасность

5.1.1 Анализ эргономики программного продукта

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

Менеджер паролей – программный продукт, разрабатываемый в рамках выпускной квалификационной работы. Назначение данного продукта является оптимизация процессов управления логинами и паролями пользователей.

Критериями оценки эргономичности программы является:

– цена ошибки (стоимость ошибки, произошедшей в результате 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, по ошибке пользователь удалил логин и пароль от рабочей системы, после чего потерял к ней доступ. Вероятно, на предприятии эти данные больше не имеют копии. Пользователю 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 с помощью самого приложения);

– средняя (данные восстанавливаются пользователем);

– высокая (данные были утеряны безвозвратно).

Интуитивность графического интерфейса напрямую связанна со скоростью обучения. Графический интерфейс не должен вводить пользователя в

заблуждение. Значки и символы, используемые в приложении, должны соответствовать предоставляемому функционалу (иконка «плюсик» отвечает за добавление, иконка «корзина» отвечает за удаление). Также необходимо учитывать следующие показатели:

– наличие одинакового структурирования пользовательского интерфейса, настроек, документации;

– отсутствие дублирования функций, настроек, программных окон и элементов управления в разных компонентах приложения.

– наличие в пользовательском интерфейсе в каждый момент времени приложения на действия пользователя.

– отсутствие окон, не имеющих смысловой нагрузки и некорректной информации.

Окно входа в приложение на компьютере изображено на рисунке 5.1. Оно содержит элементы списка, реагирующего на нажатие, и кнопки перехода на страницу регистрации.

Пользовательский интерфейс выполнен в соответствии со стилем Fluent Design, который свойственен операционной системе Windows 10. Это позволяет пользователю с большей возможностью обучиться данному программному продукту.

После запуска пользователю предоставляется список зарегистрированных в системе аккаунтов. По нажатию на соответствующий логин, пользователю открывается поле для ввода пароля. Также данное окно располагает реагирующим на нажатие текст (для перезагрузки списка пользователей и для изменения IP-адреса подключения).

В случае возникновения ошибки в результате каких-либо действий (ошибки при неверном нажатии, неверный ввод данных), пользователю показывается всплывающее окно с описание ошибки и предположительной причиной.

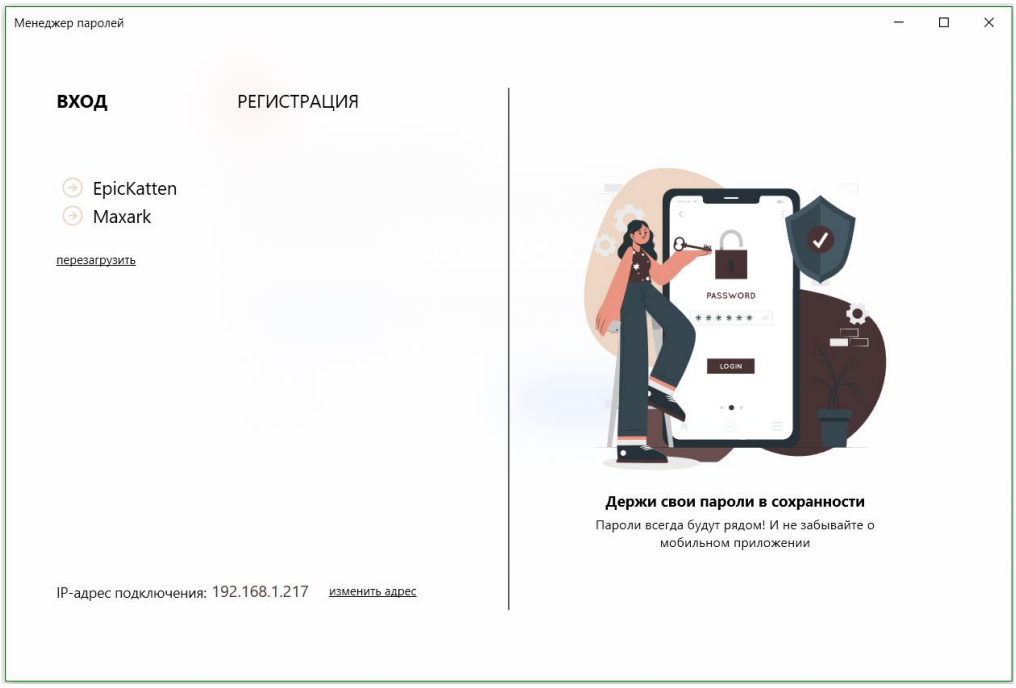

Рисунок 5.1 – Окно входа в приложение менеджер паролей

Для уменьшения стоимости ошибочного выбора аккаунта предусмотрена ссылка на ранее представленный список под полем ввода пароля. Это значит, что пользователь не сможет увидеть конфиденциальную информацию другого.

Таблица 5.1 – Анализ окна входа приложения на компьютере

Критерии	Оценка	Описание
Цена ошибки	Низкая	Запрос пароля после выбора аккаунта не позволяет другим пользователям получить доступ к конфиденциальной информации.
Понятность интерфейса	Полностью понятен	При отрисовки интерфейса использовались стандартные элементы управления (кнопки, список с выбором, подчеркнутый текст, реагирующий на нажатие). Подобные элементы используются в приложениях разного типа (мобильные, десктопные, веб-приложения).
Сложность обучения	Низкая	Все элементы интерфейса подписаны их функциональным назначением, из-за чего отсутствует двусмысленность фраз. Пользователю не составит сложности в понимании смысла каждого элемента управления, а следовательно, в освоении данной программы.

Интерфейс главного экрана приложения на компьютере представлен на рисунке 5.2. Он состоит из области управления аккаунтом (его удаление и выход из системы) и области паролей, в которой отображаются сокращенные и полные версии пароля с поиском необходимых данных. Соблюдается соответствие между интерфейсом окна на рисунке 5.1.

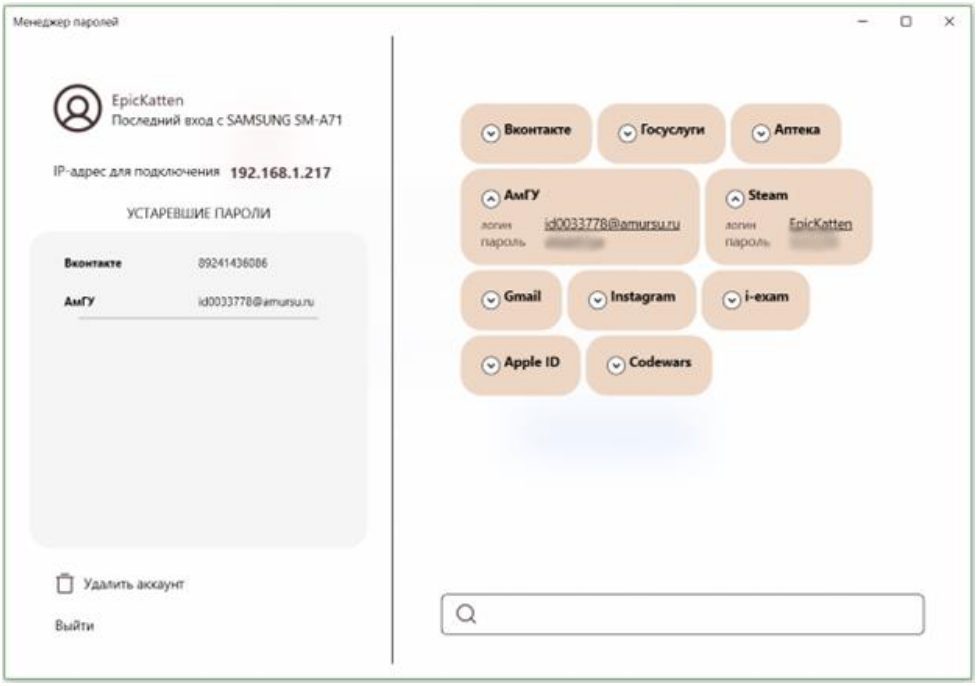

Рисунок 5.2 – Интерфейс главного окна

Удаление аккаунта способно нанести ущерб, так как пользователь потеряет все резервные копии паролей на компьютере. Для решения данной проблемы добавлено окно подтверждения удаления, где у пользователя запрашивается логин и пароль от действующего аккаунта. Запрос на подтверждение удаления аккаунта показан на рисунке 5.3. При неверном вводе данных, пользователю показывается сообщение об ошибке с описанием примерной причины.

После удаления аккаунта удаляются все резервные копии данных, поэтому после данной операции пользователя переводят на страницу, представленную на рисунке 5.1. Также на той странице будет отсутствовать удаленный логин.

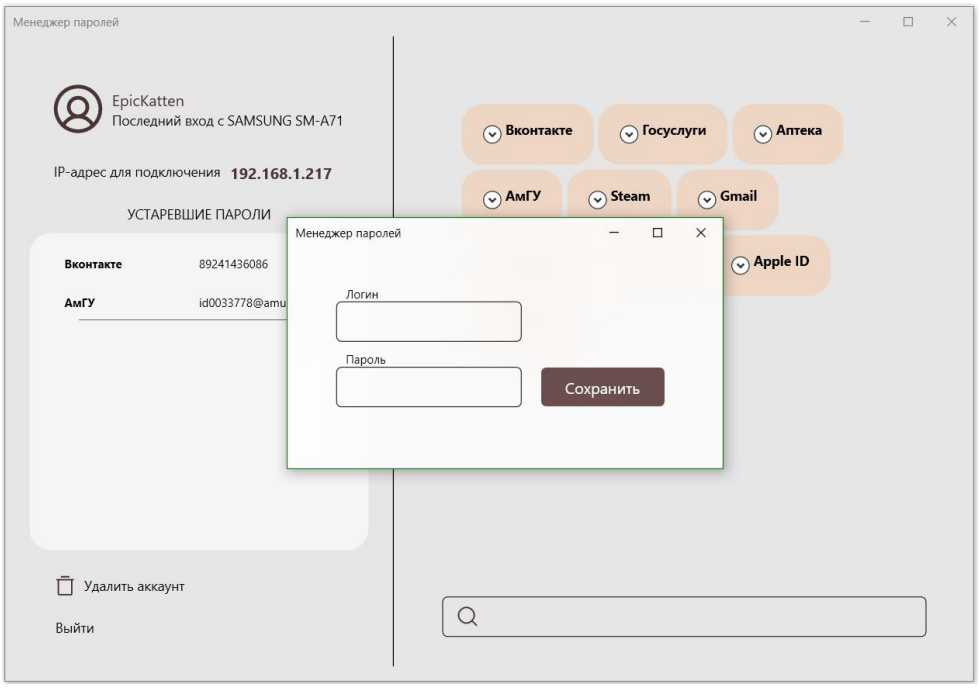

Рисунок 5.3 – Подтверждение удаления аккаунта

Таблица 5.2 – Анализ главного окна приложения на компьютере

Критерии	Оценка	Описание
Цена ошибки	Низкая	Запрос у пользователя подтверждения перед удалением. Подтверждение состоит из ввода действующих логина и пароля, чтобы избежать случайного удаления аккаунта и резервных копий на компьютере.
Понятность интерфейса	Полностью понятен	При отрисовки интерфейса использовались стандартные элементы управления (кнопки, список с выбором, подчеркнутый текст, реагирующий на нажатие). Подобные элементы используются в приложениях разного типа (мобильные, десктопные, веб-приложения).
Сложность обучения	Низкая	Все элементы интерфейса подписаны их функциональным назначением, из-за чего отсутствует двусмысленность фраз. Пользователю не составит сложности в понимании смысла каждого элемента управления, а следовательно, в освоении данной программы.

Интерфейс главной страницы приложения на мобильном устройстве представлен на рисунке 5.4. Он состоит из кнопок перехода на страницу

уведомлений и открытия меню приложения, поисковой строки и списка паролей. Из списка паролей возможны переход на страницу изменения соответствующих паролей и удаление. Чтобы удалить пароль или изменить его, необходимо открыть ниспадающих список опций (знак стрелки, направленный вниз).

В меню навигации есть соответствующая кнопка закрытия (знак крестика); список страниц, на которые возможен переход (содержит значок страниц и название); большая кнопка, которая переводит на страницу создания пароля. Интерфейс меню навигации также представлен на рисунке 5.4.

Удаление пароля на главной странице может принести ущерб, если пользователь не синхронизировал мобильное и серверное приложение. Например, возможно случайное удаление пароля от рабочей системы, к которой пользователь продолжительное время не сможет получить доступ из-за восстановления. Чтобы удалить пароль, необходимо отрыть список опций, представленный на рисунке 5.5, и выбрать пункт «Удалить».

Таблица 5.3 – Анализ главной страницы мобильного приложения

Критерии	Оценка	Описание
Цена ошибки	Средняя	Нет прямой возможности удаления пароля из системы, что уменьшает риск случайного нажатия. Но присутствует возможность случайного нажатия в опциональном списке к паролю.
Понятность интерфейса	Полностью понятен	При отрисовки интерфейса использовались стандартные элементы управления (кнопки, список с выбором, подчеркнутый текст, реагирующий на нажатие). Подобные элементы используются в приложениях разного типа (мобильные, десктопные, веб-приложения).
Сложность обучения	Низкая	Все элементы интерфейса подписаны их функциональным назначением, из-за чего отсутствует двусмысленность фраз. Пользователю не составит сложности в понимании смысла каждого элемента управления, а следовательно, в освоении данной программы.

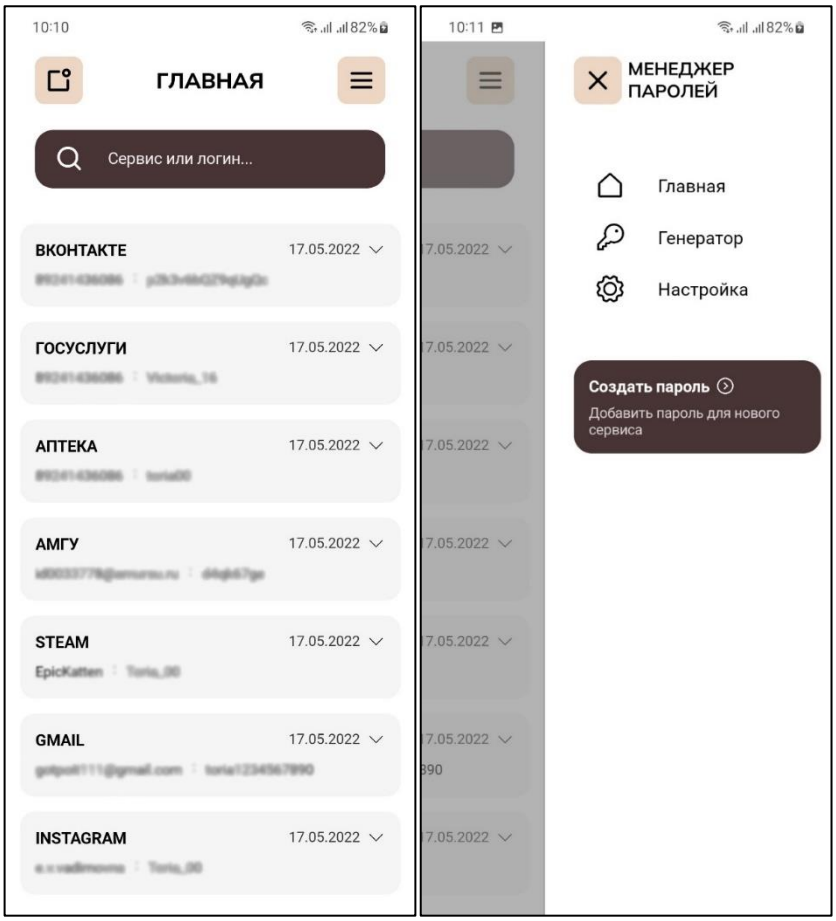

Рисунок 5.4 – Главная страница и меню навигации приложения на мобильном устройстве

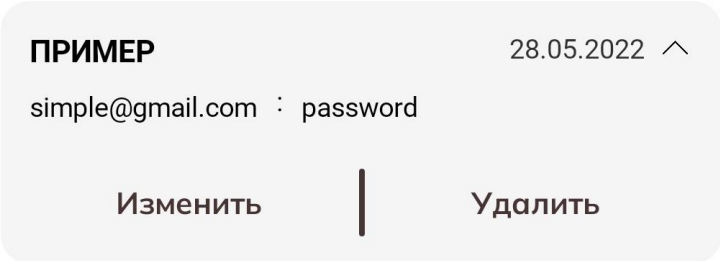

Рисунок 5.5 – Список опций (удаление и изменение пароля)

Чтобы понизить цену ошибки (со средней на низкую), необходимо запрашивать у пользователя подтверждение о удалении выбранного пароле, чтобы избежать случайных нажатий (например, когда пользователю было необходимо изменить существующий пароль, а вместо этого он удалил существующий).

Цветовая гамма интерфейса должна быть сбалансирована. Элементы управление должны иметь яркие и выделяющие цвета, если на них необходимо сделать акцент, то есть они несут наибольшую смысловую нагрузку. Шрифт, рисунки и символы не должны быть мелкими или громоздкими, в соответствии с важностью той или иной информации.

Таблица 5.4 – Размер шрифта интерфейса

Устройство	Header (Заголовок)	Title (Заглавие)	Subtitle (Подзаголовок)	Body Text (Текст)	Details (Детали)	Smallest (Малый)
Мобильное устройство	32-38 px	24-28 px	18-20px	16-18 px	13-14 px	12 px
Компьютер	40-48 px	34-38 px	21-24 px	20-21 px	16-18 px	16 px

Рассмотренные выше размеры шрифта являются рекомендованными для мобильного устройства и компьютера в рамках восприятия человеком информации.

5.1.2. Анализ опасных и вредных факторов на рабочем месте пользователя ЭВМ

Рассмотрим возможные опасные и вредные факторы на рабочем месте пользователя. Программа предоставляется в мобильной версии и версии для ЭВМ. Из этого следует, что необходимо проанализировать рабочее место пользователя ЭВМ.

Согласно пункту 6.3 СП 2.2.3670-20, рабочее место для работы в положении сидя должно соответствовать требованиям:

− пространство для размещения ног высотой не менее 600 мм;

− на уровне колен глубина составляет не менее 450 мм;

− на уровне стоп глубина составляет не менее 600 мм;

− шириной не менее 500 мм.

В СП 2.2.3670-20 определена площадь для одного рабочего места пользователя ПЭВМ с использованием плоских дискретных экранов, и она должна составлять не менее 4,5 кв.м.

Согласно СП 2.2.3670-20 (п. 251), ПЭВМ следует размещать так, чтобы показатели освещенности не превышали установленных норм, установленных в СанПиН 1.2.3685-21. Согласно ранее упомянутому стандарту, к рабочему месту предъявляются следующее требование: при площади рабочей поверхности более 0,1 кв.м. наибольшая допустимая яркость 500 кд/кв.м. Средняя освещенность для рабочего места пользователя ПЭВМ должна быть не менее 300 лк.

5.1.3 Правила безопасного пользования мобильными телефонами

Мобильное устройство, даже при развивающихся технологиях, все еще обладает вредными для людей излучениями. При близком контакте с человеком разного рода излучения способны нарушить работу органов, возможен прогрев тканей ушной раковины и головы при длительном разговоре.

Рассмотрим несколько правил для использования мобильного устройства как средства связи:

– не располагать мобильное устройство около органов, восприимчивых к излучению, то есть спереди на поясе, около головы и сердца (носить телефон в ручных или поясных сумках);

– при длительном разговоре по телефону менять положение телефона (менять ухо во время разговора), тем самым снижая нагрев тканей;

– выключать мобильное устройство в местах отсутствия сети, так как телефон находится в режиме активного поиска, из-за чего излучает большое количество излучений;

– не класть телефон рядом с головой во время сна, так как мобильное устройство излучает на частоте гамма-ритмов, схожих с ритмом мозга во сне.

Не рекомендуется также использовать мобильный телефон беременным женщинам и детям до 16 лет. Это связано с тем, что дети больше всего восприимчивы к излучениям в раннем возрасте. Также в эту рекомендацию попадают люди с таким заболеваниями и расстройствами, как: эпилепсия (излучения провоцирую припадок), нервные заболевания, сниженная умственная и физическая активность, расстройство памяти и сна.

Причиной излишне сильных излучений у современных мобильных устройств состоит в том, что антенна располагается внутри, повышая мощность излучений. У телефонов с внешней антенной излучений значительно меньше.

Далее рассмотрим влияние мобильного устройства на зрение человека. Врачи придерживаются мнения, что глаза подвержены меньшему напряжению, если пользоваться мобильным телефоном на расстоянии более 40 см и ниже уровня глаз. Примерное положение устройства относительно глаз показано на рисунке 5.6. При расстоянии менее 25 см у большинства людей возникают признаки близорукости.

Рисунок 5.6 – Положение устройства относительно глаз

Перечислим необходимые мероприятия по предотвращению появления глазных заболеваний и ухудшения зрения;

Моргать чаще. При активном использовании устройства, человек моргает в три раза меньше, что приводит к сухости глаз.

Осуществлять отдых для глаз. Через каждые 20 минут использования устройства, необходимо переводить взгляд на что-то отдаленное хотя бы на минуту.

Не использовать телефон в темном помещении. Недостаток освещения при яркой подсветке устройства вредят глазам.

Проводить гимнастику для глаз. Гимнастика состоит в фокусировки зрения на объектах, находящихся на разном расстоянии. Данное упражнение необходимо проводить около 5 минут.

Проверять зрение у специалиста. Регулярная проверка зрения способна предупредить некоторые болезни и следить за зрением.

5.1.4 Правила работ за компьютером

Помимо вредных воздействий от мобильного устройства существуют также воздействия от монитора и компьютера в целом. При неправильном положении возможны такие проблемы со здоровьем, как искривление позвоночника (и болезни, следующие от этого) и проблемы со зрением. На рисунке 5.7 показаны положения, при котором необходимо работать с компьютером.

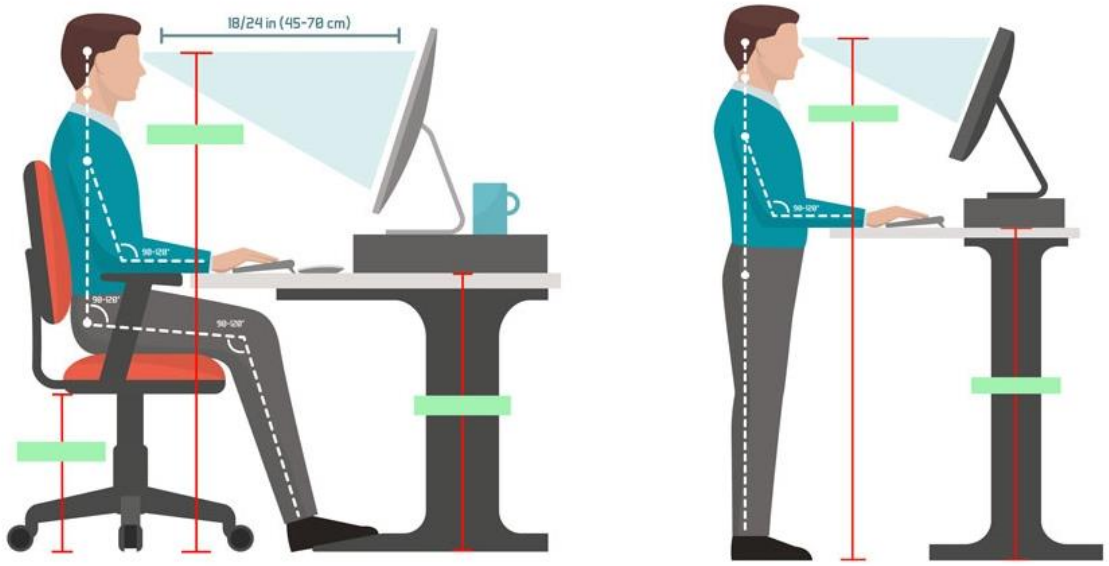

Рисунок 5.7 – Положения человека при работе с компьютером

Далее рассмотрим правила для комфортной работы с компьютером:

Соблюдать дистанции между монитором и глазами. Необходимо располагать монитор на расстоянии 70 см от глаз. Клавиатуру и экран нужно держать прямо перед собой, чтобы избежать поворотов.

Избегать напряжения тела. Голову необходимо держать прямо, руки располагать прямо на клавиатуре, чтобы запястья были расслабленными. Спинка стула может быть слегка откинутой, но спина при этом должна оставаться ровной.

Делать периодический отдых. Долго сидение на одном месте вредит здоровью, поэтому необходимо в среднем раз в час делать перерыв от работы, делать гимнастику несколько раз пройтись по помещению. Помимо этого, необходим отдых для глаз. Для этого поможет фокусировка на предметах, расположенных на разных расстояниях. Данное упражнение поможет снять напряжение с глаз.

Данные мероприятия необходимо выполнять как при работе с компьютером на предприятии, так и при работе в домашних условиях. Нужно делать периодические перерывы, делать гимнастику или простой обход помещений, чтобы снять напряжение с тела.

Не стоит также забывать о моргании во время работы с компьютером, потому что он, как и мобильное устройство, вызывает сухость глаз.

5.2 Экологичность

Отходы производства – вещества или предметы, образованные в процессе производства, которые предназначены для удаления или подлежат удалению. Утилизация отходов – использование отходов для производства продукции и выполнения работ, повторное применение отходов, их возврат в производственный цикл после соответствующей подготовки, а также извлечение полезных компонентов для их повторного применения. Утилизация отходов является главной задачей, которая стоит перед предприятием для обеспечения экологической безопасности окружающей среды. Организация и управление утилизацией отходов являются весьма сложным делом. Заниматься этой деятельностью должны специально подготовленные люди.

Организация должна обеспечить уничтожение мусора, соответствующее необходимым требованиям и стандартам. Несоблюдение данных норм способно повлечь серьезные проблемы для предприятия.

Для организации утилизации необходимо применять различное оборудование, так как различного рода материл, утилизируется отдельно друг от друга (например, строительные отходы утилизируются отдельно от химических).

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

Организация, производящая какой-либо продукт, производит следующие отходы (указанные в ГОСТ 30772-2001):

− вторичная продукция – вещества, материалы, комплектующие изделия, детали, функциональные узлы, блоки, агрегаты от различных объектов, утратившие свои потребительские свойства и не пригодные для дальнейшей эксплуатации в соответствии с директивными требованиями и/или нормативной документацией, но представляющие собой товарную продукцию.

− отходы производства – 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. Рассмотрим варианты утилизации данной продукции.

Возможна *переработка объектов*, состоящих преимущественно из пластмассы. Однако существуют такие объекты микроэлектроники, которые не подлежат переработке. Такие объекты отправляются на вторичное использование.

Возможна *продажа вторичной* продукции как товаров бывших в употреблении. Тогда для организации появляется возможность окупить некоторые объекты производства, не затрачивая ресурсов на утилизацию.

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. Также к объектам для утилизации относятся такие вещи, как бумага и мебель, мелкие компоненты микроэлектроники, которые не подлежат повторному использованию.

5.3 Чрезвычайные ситуации

Наиболее частой чрезвычайной ситуацией в организации является пожар. Причинами такому являются: оставленные без присмотра электроприборы, отклонение от техники безопасности и правил пожарной безопасности, курение в неположенных местах и тд.

Согласно НПБ 105-03, помещения с ЭВМ являются пожароопасными в категории В1 – В4, в которых могут содержатся материалы, способные гореть при взаимодействии с водой или друг другом. Путь эвакуации не должен быть захламлен или заставлен мебелью или техническими приборами. Провода не должны располагаться на полу.

Для предотвращения возможного пожара необходимо соблюдать следующие правила:

– не хранить горючие жидкости, взрывчатые вещества рядом с ЭВМ, не использовать из рядом с ЭВМ;

– не использовать электронагревательные приборы;

– не использовать провода с поврежденной изоляцией;

– не использовать поврежденные розетки и электроприборы;

– не хранить бумагу рядом с нагревательными элементами и электроприборами, склонных к нагреву;

– не курить в помещении с ЭВМ;

– не оставлять без наблюдения включенные в сеть электроприборы;

– не ремонтировать блоки ЭВМ в помещениях их расположения;

– не нарушать правила эксплуатации ПЭВМ;

– раз в 3 месяца необходимо проводить санитарную очистку рабочих мест и ПЭВМ.

По завершении работы необходимо отключить от сети электроприборы, зафиксировать отсутствие возгорания. При возникновении возгорания,

необходимо позвонить в пожарную службу, сообщить всю необходимую информацию, подготовить к эвакуации материальные ценности, документацию и покинуть здание через запасные выходы.

ЗАКЛЮЧЕНИЕ

На сегодняшний день насчитывается большое количество сервисов, в том числе и мобильных, к которым необходим доступ. В целях безопасности, для каждого веб-сервиса необходимы свои неповторяющиеся и надежные аутентификационные данные. В таком случае встает нужна в безопасном хранении этих данных в пределах доступности пользователя.

В ходе выполнения работы был произведен анализ существующих аналогов мобильного менеджера паролей, в результате которого были выявлены некоторые недостатки этих систем, таких как высокая стоимость некоторых функций, отсутствие резервных копий на локальных устройствах и переносных хранилищах. На основании выявленных недостатков были определены требования к системе и методология проектирования, по которой разрабатывалось приложение.

Произведено проектирование баз данных, которые включают одну таблицу бля мобильного устройства и две для компьютера. Представлена возможность внедрения новых таблиц при включении в приложения пользовательских категорий. Построены логические и физические модели с помощью CASE-средства ERwin Data Modeler. На основании требований к проекту выявлены и спроектированы функциональные модели (управления аккаунтом, управления паролями, генерации паролей, клиент-серверного взаимодействия). Схемы построены в Microsoft Visio 2019.

На основании выбранных архитектур (клиент-серверный и MVVM шаблоны) и средств разработки (C# и XAML; Xamarin, WPF и ASP NET Core; SQLite; Visual Studio 2022) было разработано полностью готовое приложение «Менеджер паролей» с функцией синхронизации на компьютере. Приведены скриншоты работы приложений как общей системы.

Произведена оценка информационной безопасности, которая обеспечивается в разработанном приложении. Мерами обеспечения безопасности в приложении являются шифрование пользовательских данных алгоритмом

AES, хэширование паролей для входа в систему стандартом RFC 2898. Данные передаются по локальной сети по протоколу HTTPS. Оценена безопасность жизнедеятельности, проверены возможные ошибки при использовании приложения и приведены рекомендации по их устранению.

Готовое мобильное приложение прошло регистрацию, выходным документом которой является свидетельство о государственной регистрации программы для ЭВМ. Название: Менеджер паролей для смартфона. Свидетельство представлено в приложении Б.

На данный момент изучается материал, который позволит добавить в приложение такие функции, как: автозаполнение электронных полей, встроенная аутентификация и двухфакторная аутентификация.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Булгаков, А. Б. Безопасность жизнедеятельности [Электронный ресурс] : сб. учеб.-метод. материалов / АмГУ, ИФФ ; сост.: А. Б. Булгаков, В. Н. Аверьянов, М. В. Гриценко. – Благовещенск : Изд-во Амур. гос. ун-та, 2017. – 176 с. – Б. ц.

2 Гвоздева, Т. В. Проектирование информационных систем/ Т. В. Гвоздева. – М.: Феникс, 2009. – 512 с.

3 Грекул, В. И. Проектирование информационных систем/ В. И. Грекул. - М.: Интернет-Университет Информ.технологий, 2005.- 304 с.

4 Гринберг, С. UX-дизайн = Sketching user experiences : идея - эскиз - воплощение / С. Гринберг [и др.] ; [пер. с англ. Е. Карманова]. – Санкт-Петербург : Питер, 2014. – 272 с. : ил. - (Библиотека специалиста). - Библиогр. в конце разд. - ISBN 978-5-4461-0203-7 : 444-00.

5 Дейт, К.Дж. Введение в системы базы данных / К. Дж. Дейт – Вильямс, 2001. – 1072 с.

6 Кардаш, Т. А. Эргономика рабочих мест служащих и инженерно-технических работников, оснащенных ПЭВМ [Текст] : учеб. пособие / Т. А. Кардаш ; АмГУ, ИФФ. – Благовещенск : Изд-во Амур. гос. ун-та, 2002. – 60 с.

7 Коровкина, Н.Л. Проектирование ИС/ Н.Л. Коровкина. - . - М.: Феникс, 2007.- 305 с.

8 Маклаков С.В. Создание информационных систем с AllFusion Modeling Suite / С.В. Маклаков. –М.: Диалог-Мифи, 2003. – 432 с.

9 Олифер, В. Г. Основы компьютерных сетей : [учеб. пособие для вузов] / В. Г. Олифер, Н. А. Олифер. – Санкт-Петербург : Питер, 2014. – 400 с. : ил.

10 Павловская, Т. А. C#. Программирование на языке высокого уровня : [учеб. для вузов] / Т. А. Павловская. – Санкт-Петербург : Питер, 2014. – 432 с. : ил. – (Учебник для вузов). – Библиогр.: с. 425-426. – ISBN 978-5-496-00861-7 : 330-00.

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

12 Тузовский, А.Ф. 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: 2.06.2022).

13 Шумилин, В. К. Пособие по безопасной работе на персональных компьютерах [Текст] / разраб. Шумилин. – М. : НЦ ЭНАС, 2005. – 28 с.

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

15 Fluent Design System [Электронный ресурс] // URL: https://www.microsoft.com/design/fluent/#/

16 Habr: Fluent Design (не) сдвигая парадигмы [Электронный ресурс] // URL: https://habr.com/ru/post/329106/

17 Habr: MVVM: полное понимание (+WPF) Часть 1 [Электронный ресурс] // URL : https://habr.com/ru/post/338518/

18 Habr: Подробно о Xamarin [Электронный ресурс] // URL: https://habr.com/ru/post/188130/

19 METANIT.COM, Сайт о программировании [Электронный ресурс] // URL : https://metanit.com/

10 Microsoft Docs, Microsoft technical documentation [Электронный ресурс] // URL : https://docs.microsoft.com/

ПРИЛОЖЕНИЕ А

Техническое задание на разработку программного обеспечения

1. Введение

1.1. Наименование программы

Полное наименование: Менеджер паролей

Условное обозначение: YourPasswords

1.2. Краткая характеристика области применения

«Менеджер паролей» – программа, осуществляющая безопасное хранение и управление аутентификационными данными пользователей системы; генерацию надежных паролей для входа в различные веб-сервисы; резервное копирование данных на компьютер и отображение этих данных в приложении на компьютере.

Цель разрабатываемой программы: упростить и обезопасить хранение конфиденциальных данных пользователей.

2. Основания для разработки

2.1. Документы, на основании которых ведется разработка

На основании приказа об утверждении темы бакалаврской работы.

2.2. Наименование и/или условное обозначение темы разработки

Наименование темы: Разработка мобильного приложения «Менеджер паролей» с синхронизацией на компьютере.

3. Назначение разработки

3.1. Функциональное и эксплуатационное назначение

Функциональное назначение: Приложение создает, хранит, изменяет и удаляет (осуществляет управление) аутентификационные данные пользователей системы. Приложение генерирует надежные пароли с отображением примерного времени перебора пароля методом грубой силы. Помимо этого, приложение осуществляет резервное копирование данных на компьютер по запросу пользователя.

Эксплуатационное назначение: Приложение состоит из трех компонентов – клиент на мобильном устройстве, клиент на компьютере, серверное приложение. Мобильный клиент является самостоятельным приложением, то есть остальные компоненты являются опциональными. В свою очередь, клиент на компьютере напрямую зависит от серверного приложения.

4. Требования к программе

4.1. Требования функциональным характеристикам

Программа состоит из трех основных компонентов: клиенткой части на компьютере, серверной части и клиентской части на компьютере, между которыми должно быть налажено взаимодействие.

4.1.1. Требования к серверной части

В серверной части должно быть реализовано шифрование аутентификационных данных пользователя при сохранении их в базу данных. При возможности файл с базой данных должен скрываться от пользователя во избежание случайных открытий.

Помимо шифрование конфиденциальных данных, серверная часть должна обеспечивать хэширование с добавлением соли пароля для входа в систему. Далее этот пароль используется в качестве ключа (то есть становится мастер-паролем), переводя его к размеру в 256 бит.

4.1.2. Требования к взаимодействию клиенткой и серверной части

Взаимодействие между серверной и клиентскими частями должно осуществляться за счет HTTP-запросов (GET, POST, PATCH, DELETE). Данные передаются в виде файлов JSON, который имеет структуру данных «ключ-значение».

4.1.3. Требования к клиентским частям

Мобильная клиентская часть реализовывается в виде приложения на мобильное устройство. Ее главными функциями является создание, хранение, изменение и удаление паролей. При этом главные аутентификационные данные (логин и пароль) должны храниться в мобильной базе данных в зашифрованном виде (то есть наличие метода шифрования).

Также на мобильном устройстве должна быть осуществима синхронизация данных, то есть отправка резервных копий данных на серверное приложение. Для этого должна быть реализована авторизация на мобильном устройстве. Как только пользователь авторизовывается в системе, на сервер отправляется модель и название устройства, с которого произошел вход.

Клиентская часть на компьютере осуществляет управление аккаунтами (добавление, удаление, вход в систему). Помимо этого, пользователю предоставляется список хранящихся паролей на серверном приложении.

4.2. Требования к надежности

4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы

Для автономной работы мобильного приложения не предъявляются никаких требований к надежности и устойчивому функционированию. При отправке данных на сервер обязательно должно быть включено серверное приложение, мобильное устройство должно быть подключено к локальной сети, в которой находится серверное приложение.

Для работы клиентского приложения на компьютере необходимо постоянное подключение к серверу и локальной сети. При возможности, эти части приложения должны быть установлены на одном устройстве.

4.2.2. Время восстановления после отказа

В случае отказа работы одной из частей системы, необходимо перезагрузить отказавшую часть. Время восстановления – время перезагрузки.

4.2.3. Отказы из-за некорректных действий оператора

При отказе из-за некорректных действий оператора должно быть исключено возможное отключение программы. Оператору должно быть предоставлено сообщение об ошибке с предположительной причиной.

4.3. Условия эксплуатации

4.3.1. Климатические условия эксплуатации

Требования к климатическим условиям эксплуатации не предъявляется.

4.3.2. Требования к видам обслуживания

Обслуживание не требуется.

4.3.3. Требования к численности и квалификации персонала

Для управления мобильным приложением и приложением на компьютере достаточно базовых знаний работы мобильного устройства и компьютера. Для включения серверного приложения достаточно знания строки подключения (который предоставляется в руководстве пользователя).

4.4. Требования к составу и параметрам технических средств

Минимальные требования для мобильной устройства:

− операционная система: Android 8 и выше, iOS 10 и выше;

− оперативная память: 1 ГБ;

− занимаемая память: 50 МБ.

Доступ к локальной сети при необходимости резервного копирования.

Минимальные требования для компьютера имеют вид:

− операционная система: Windows 10 и выше;

− оперативная память: 100 МБ;

− занимаемая память: 80 МБ.

Обязательное наличие локальной сети для подключения мобильного устройства.

4.4. Требования к информационной и программной совместимости

4.4.1. Требования к исходным кодам и языкам программирования

Исходные коды приложения должны быть написаны на языке C# и XAML.

4.4.2. Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы не ниже Windows 10. На системе должен быть установлен .NET Framework 4.5.

4.5. Требования к маркировке и упаковке

Требований к маркировке и упаковке не предъявляются.

4.6. Требования к транспортировке и хранению

Мобильное приложение может распространяться в магазинах мобильных приложений, таких как Google Play и App Store. Приложение для компьютера может распространяться в магазине приложений Microsoft Store.

5. Требования к программной документации

5.1. Состав программной документации

− «Менеджер паролей». Техническое задание (ГОСТ 19.201-78);

− «Разработка мобильного приложения «Менеджер паролей» с синхронизацией на компьютере. Текст бакалаврской работы.

6. Технико-экономические показатели

6.1. Ориентировочная экономическая эффективность

В рамках данной работы расчет экономической эффективности не предусмотрен.

6.2. Предполагаемая потребность

Потребность в разработанной системе обуславливается в хранении аутентификационных данных пользователей от различных сервисов, включая мобильных. Также предусмотрена потребность в безопасности, то есть резервные копии хранятся на домашнем компьютере пользователя, а не на глобальном сервере.

6.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

Существующие аналоги обладают только глобальным сервером, то есть данные хранятся только на глобальном сервере, а следовательно, могут быть подвержены атакам нарушителей. Помимо этого, аналоги предоставляются либо с платным тарифом, либо с ограниченным числом функций, что ограничивает работу пользователя.

7. Стадии и этапы разработки

7.1. Необходимые стадии разработки, этапы и содержание работ

Этапы жизненного цикла разработки:

− сбор требований (сбор, систематизация требований к системе);

− проектирование (выбор языка программирования, среды разработки системы; выбор архитектуры);

− кодирование (разработка системы);

− тестирование (проверка на соответствие требованиям);

− разворачивание (передача системы в магазины приложений).

7.2. Сроки разработки и исполнители

Сроки разработки: нет точно установленных сроков.

Исполнитель: Евдокимова Виктория Вадимовна.

ПРИЛОЖЕНИЕ Б

Свидетельство о государственной регистрации программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022619581

Менеджер паролей для смартфона

Правообладатель: ***Федеральное государственное бюджетное образовательное учреждение высшего образования «Амурский государственный университет» (RU)***

Авторы: ***Галаган Татьяна Алексеевна (RU), Евдокимова Виктория Вадимовна (RU)***

Заявка № **2022618449**
Дата поступления **06 мая 2022 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***24 мая 2022 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 68b800778e14e40f0a94edbd24145d5c7
Владелец Зубов Юрий Сергеевич
Действителен с 26.05.2022 по 26.05.2023

Ю.С. Зубов

ПРИЛОЖЕНИЕ В

Шифрование и дешифрование данных

```
32      public static byte[] EncryptData(string data)
33      {
34          byte[] ClearData = Encoding.UTF8.GetBytes(data);
35
36          SymmetricAlgorithm Algorithm = SymmetricAlgorithm.Create(AlgorithmName);
37          Algorithm.Key = Convert.FromBase64String(Key);
38
39          MemoryStream Target = new MemoryStream();
40
41          Algorithm.GenerateIV();
42          Target.Write(Algorithm.IV, 0, Algorithm.IV.Length);
43
44          CryptoStream cs = new CryptoStream(Target, Algorithm.CreateEncryptor(), CryptoStreamMode.Write);
45          cs.Write(ClearData, 0, ClearData.Length);
46          cs.FlushFinalBlock();
47
48          return Target.ToArray();
49      }
```

Рисунок В.1 – Шифрование данных

```
51      public static string DecryptData(byte[] data)
52      {
53          SymmetricAlgorithm Algorithm = SymmetricAlgorithm.Create(AlgorithmName);
54          Algorithm.Key = Convert.FromBase64String(Key);
55
56          MemoryStream Target = new MemoryStream();
57
58          int ReadPos = 0;
59          byte[] IV = new byte[Algorithm.IV.Length];
60          Array.Copy(data, IV, IV.Length);
61          Algorithm.IV = IV;
62          ReadPos += Algorithm.IV.Length;
63
64          CryptoStream cs = new CryptoStream(Target, Algorithm.CreateDecryptor(),
65              CryptoStreamMode.Write);
66          cs.Write(data, ReadPos, data.Length - ReadPos);
67          cs.FlushFinalBlock();
68
69          return Encoding.UTF8.GetString(Target.ToArray());
70      }
```

Рисунок В.2 – Дешифрование данных

ПРИЛОЖЕНИЕ Г

Хэширование пароля и проверка на подлинность

```
8   public static string HashPassword(this string password)
9   {
10      byte[] salt;
11      byte[] buffer2;
12      using (Rfc2898DeriveBytes bytes = new Rfc2898DeriveBytes(password, 0x10, 0x3e8))
13      {
14          salt = bytes.Salt;
15          buffer2 = bytes.GetBytes(0x20);
16      }
17      byte[] dst = new byte[0x31];
18      Buffer.BlockCopy(salt, 0, dst, 1, 0x10);
19      Buffer.BlockCopy(buffer2, 0, dst, 0x11, 0x20);
20      return Convert.ToBase64String(dst);
21  }
```

Рисунок Г.1 – Хэширование пароля

```
23  public static bool VerifyHashedPassword(this string hashedPassword, string password)
24  {
25      byte[] buffer4;
26      if (hashedPassword == null)
27      {
28          return false;
29      }
30      byte[] src = Convert.FromBase64String(hashedPassword);
31      if ((src.Length != 0x31) || (src[0] != 0))
32      {
33          return false;
34      }
35      byte[] dst = new byte[0x10];
36      Buffer.BlockCopy(src, 1, dst, 0, 0x10);
37      byte[] buffer3 = new byte[0x20];
38      Buffer.BlockCopy(src, 0x11, buffer3, 0, 0x20);
39      using (Rfc2898DeriveBytes bytes = new Rfc2898DeriveBytes(password, dst, 0x3e8))
40      {
41          buffer4 = bytes.GetBytes(0x20);
42      }
43      return ByteArraysEqual(buffer3, buffer4);
44  }
45
46  private static bool ByteArraysEqual(byte[] b1, byte[] b2)
47  {
48      if (b1 == b2) return true;
49      if (b1 == null || b2 == null) return false;
50      if (b1.Length != b2.Length) return false;
51      for (int i = 0; i < b1.Length; i++)
52      {
53          if (b1[i] != b2[i]) return false;
54      }
55      return true;
56  }
```

Рисунок Г.2 – Проверка на подлинность введенного пароля