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

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем
Направление подготовки 09.04.04 – Программная инженерия
Направленность (профиль) образовательной программы Управление разработкой программного обеспечения

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой
_____ А.В. Бушманов
«_____»_____2024 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему: Разработка приложения для визуальной идентификации пользователя «Faseblock»

Исполнитель
студент группы 2105-ом _____ (подпись, дата) А.А. Барсук

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

Руководитель научной
магистерской программы,
профессор, доктор техн. наук _____ (подпись, дата) И.Е. Ерёмин

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

Рецензент
_____ (подпись, дата) А.А. Тодосейчук

Благовещенск 2024

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

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ
Зав. кафедрой
_____ А.В. Бушманов
«_____»_____2024 г.

З А Д А Н И Е

К магистерской диссертации студента группы 2105-ом Барсук Алёны Алексеевны

1. Тема магистерской диссертации: Разработка приложения для визуальной идентификации пользователя «Faseblock»

(Утверждено приказом от 06.03.2024 № 632-уч)

2. Срок сдачи студентом законченной работы (проекта): 10.06.2024 г.

3. Исходные данные к магистерской диссертации: отчёт по преддипломной практике

4. Содержание магистерской диссертации: анализ предметной области, алгоритм решения задачи с использованием компьютерных технологий, а также программная реализация решения.

5. Перечень материалов приложения: техническое задание

6. Рецензент магистерской диссертации: Тодосейчук Александр Александрович

7. Дата выдачи задания: 29.01.2024 г.

8. Руководитель выпускной квалификационной работы: Галаган Татьяна Алексеевна доцент, канд. техн. наук

(фамилия, имя, отчество, должность, ученая степень, ученое звание)

9. Задание принял к исполнению (29.01.2024): _____ Барсук А.А.

РЕФЕРАТ

Магистерская диссертация содержит 66 с., 22 рисунка, 2 таблицы, 50 источников.

РАСПОЗНАВАНИЕ ЛИЦ, ИДЕНТИФИКАЦИЯ, АУДИТ, УЧЕТНАЯ ЗАПИСЬ, НЕРЕЛЯЦИОННАЯ БАЗА ДАННЫХ, УДАЛЕННЫЙ ДОСТУП.

В магистерской диссертации проанализировано зарубежное программное обеспечение, на основе которого разработано приложение для идентификации пользователя, что отличается от проанализированных аналогов более широким функционалом.

Цель магистерской диссертации – создание приложения для идентификации пользователя во время его работы на персональном компьютере или ноутбуке.

Для достижения поставленной цели в рамках магистерской диссертации решаются следующие основные задачи:

- анализ методов идентификации пользователя;
- анализ существующих программных решений;
- анализ языков программирования;
- выбор средств для разработки;
- разработка приложения для персонального компьютера или ноутбука;
- разработка Telegram-бота и его внедрение в приложение.

Практической значимостью приложения является повышение эффективности активной безопасной среды каждого персонального рабочего места.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В настоящее время невозможно представить, что когда-то люди жили без интернета или мобильных телефонов. Ещё пол века назад компьютер размером с комнату был самым передовым, а телефон, что работает без проводной сети можно было встретить только в трудах фантастов. Но, с тех пор, технологии сделали качественный скачок вперёд, как по внутреннему устройству, так и по возможностям. Если раньше, даже обычный телефон был привилегией, то теперь, встретить кого-то без сотового практически невозможно, а стационарный персональный компьютер или ноутбук есть дома у каждого второго.

Наряду с развитием технологий, также прогрессировали и способы хранения данных, так с перфокарт и магнитных кассет эволюция дошла до флешек и облачных хранилищ. Однако, вместе со способами хранения, не стояли на месте и технологии их хищения. Различные компьютерные вирусы эволюционируют день ото дня, не давая заскучать специалистам, то разрабатывают антивирусное ПО. Но, данные можно похитить и не используя программы – так называемый «человеческий фактор» справляется с этим ничуть не хуже, оставляя, например, на самом видном месте записку с паролем от ноутбука, либо же, не выходя из учётной записи при использовании компьютера в интернет-кафе, оставляя свою страничку в соцсети на всеобщее обозрение.

Не редки случаи, когда пользователь оставляет на столе бумажный носитель с важными паролями, либо, уходя, не выключает устройство, на котором находится информация, необходимая злоумышленнику. Для подобных случаев также есть программы, которые позволяют предотвратить хищение информации с устройства сторонними лицами. И, если на многих сотовых телефонах, помимо сканера отпечатка пальца или обычного пароля, может стоять распознавание по фотографии лица пользователя, то на некоторых ноутбуках, а также многих стационарных персональных компьютерах , такой возможности нет. Программы, которые позволяют распознавать пользователя по его идентификатору, нуждаются в определённых сенсорах, либо же в камере, будь она встро-

енной или внешней. Однако, многие из программ, что распознают пользователя не являются отечественной разработкой, а также редко когда полностью бесплатны. В рамках выполнения магистерской диссертации проанализированы современные технологии распознавания лиц, а также проведён сравнительный анализ существующих решений для создания собственной программы по идентификации пользователя во время работы с персональным компьютером или ноутбуком.

Необходимость приложения обусловлена:

– потребностью в защите персональных данных пользователя;

– потребностью в уровнях доступа к устройствам;

– непрерывное отслеживание работы пользователя (аналог аудита).

Целью магистерской диссертации является создание приложения для идентификации пользователя во время его работы на персональном компьютере или ноутбуке, которое будет блокировать доступ нарушителю.

Для достижения поставленной цели в рамках магистерской диссертации решаются следующие основные задачи:

- анализ методов идентификации пользователя;
- анализ существующих программных решений;
- анализ языков программирования;
- выбор средств для разработки;
- разработка приложения для персонального компьютера или ноутбука;
- разработка Telegram-бота и его внедрение в приложение.

Практической значимостью приложения является повышение эффективности активной безопасной среды каждого персонального рабочего места.

Результаты работы апробированы на конференциях: XXIV региональная научно-практическая конференция «Молодежь XXI века: шаг в будущее» (18 мая 2023 г., Благовещенск), международная научно-практическая конференция «Планирование, проведение и толкование итогов научных исследований» (20 января 2024 г., г. Киров).

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Распознавание лиц – это категория биометрических систем аутентификации.

Технология распознавания лиц работает следующим образом:

- Сначала происходит процедура обнаружения лица. Камера обнаруживает и фиксирует положение изображения лица, как одного, так и в толпе.

- Затем выполняется снимок и проводится анализ изображения лица. Большинство технологий распознавания лиц используют 2D-изображения, поскольку 2D-изображения удобнее сопоставлять с общедоступными фотографиями или фотографиями в базе данных.

Современные методы распознавания лиц используют передовые технологии компьютерного зрения и машинного обучения, часто опираясь на обученные встроенные библиотеки. Вот несколько ключевых подходов и библиотек, которые применяются в этой области.

1.1 Методы распознавания лиц

1.1.1 Примитивы Хаара

Примитивы Хаара – метод извлечения характеристик изображений, который был разработан Полом Виолой и Майклом Джонсом в 2001 году. Они применили этот метод для решения проблемы обнаружения лиц, но с тех пор применяются и в других областях компьютерного зрения и обработки изображений. Примитивы Хаара являются простым, но эффективным способом описания текстур и форм на изображении.

Основные идеи примитивов Хаара основаны на использовании прямоугольных фильтров различного размера и формы для выделения различных характеристик изображения, таких как края, границы и текстуры. Примитивы Хаара выражаются в виде разнообразных шаблонов, которые применяются к каждому пикселю изображения. Эти шаблоны имеют разные весовые коэффициенты, которые определяют их вклад в вычисление характеристик (рисунок 1).

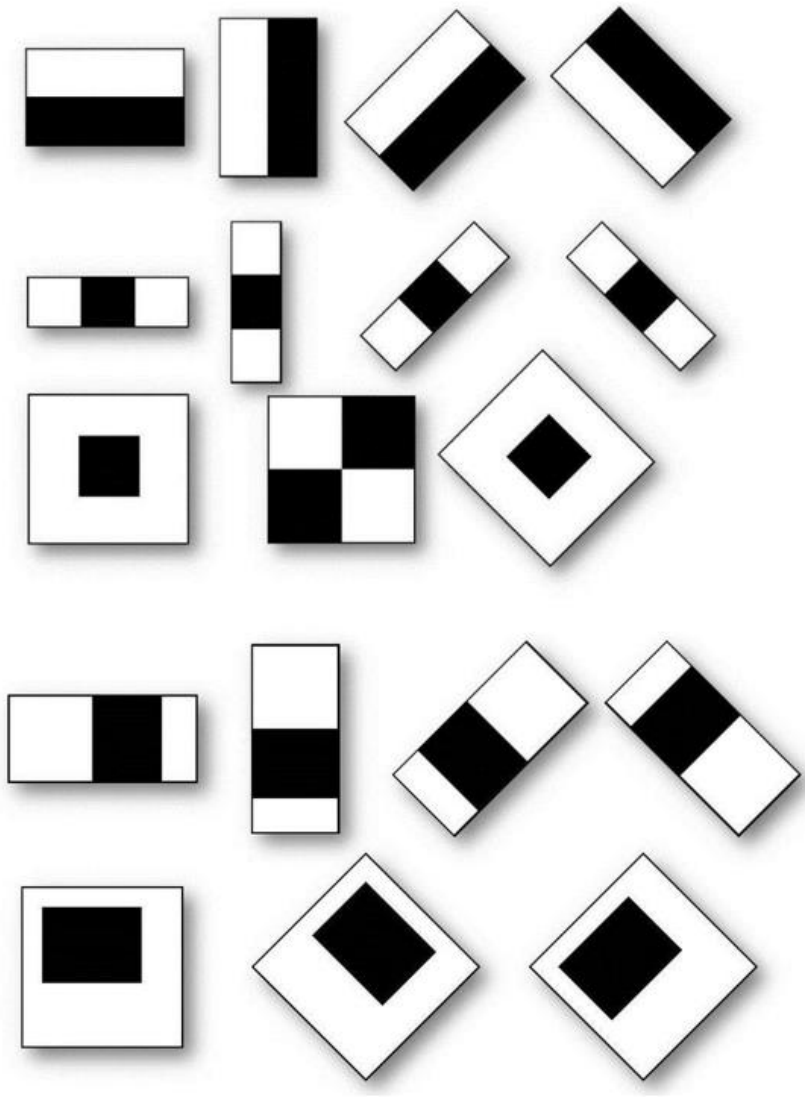

Рисунок 1 – Примитивы Хаара

Примитивы Хаара используются в каскадных классификаторах Виолы-Джонса для быстрого и эффективного обнаружения объектов на изображениях, таких как лица. Каскадный классификатор представляет собой систему последовательно применяемых классификаторов, каждый из которых использует набор примитивов Хаара для определения наличия или отсутствия объекта на изображении. Такой подход позволяет существенно ускорить процесс обнаружения объектов и снизить вычислительную нагрузку.

Также, примитивы широко используются в других задачах обработки изображений, таких как распознавание жестов, детектирование движения и сегментация изображений. Их простота и эффективность делают их популярным инструментом в области компьютерного зрения.

1.1.2 Гистограмма ориентированных градиентов (HOG)

Метод HOG используется для обнаружения лиц в изображениях. Он выделяет контуры и текстуры объекта, что позволяет определить границы лица.

Преимущества – быстрота и относительная простота вычислений.

Недостатки – выступает в точности более современным методам, особенно в сложных условиях (повороты лица, плохое освещение).

1.1.3 Методы на основе нейронных сетей

Свёрточные нейронные сети (CNN) – эти сети отлично подходят для задач распознавания лиц благодаря своей способности выделять сложные паттерны в изображениях.

Функции потерь (Loss Functions) – используются специфические функции потерь, такие как Contrastive Loss и Triplet Loss, которые помогают улучшить различимость лиц.

Архитектура VGG-Face – одна из ранних и популярных архитектур для распознавания лиц.

Архитектура FaceNet – разработана Google, использует Triplet Loss для обучения и достижения высоких показателей точности.

Архитектура DeepFace – разработана Facebook, включает несколько свёрточных слоев и используется для анализа лиц в социальных сетях.

1.1.4 Встроенные библиотеки и инструменты

Библиотека dlib с открытым исходным кодом, включающая мощные алгоритмы для распознавания лиц. Позволяет обнаружить лицо с использованием метода HOG, проводит распознавание с помощью векторизации лиц (Face Encodings), имеется поддержка нейронных сетей.

Библиотека OpenCV – использует функцию компьютерного зрения с обширным набором инструментов для обработки изображений и видео. Включает каскады (примитивы) Хаара для обнаружения лиц, а также поддерживает интеграцию с глубокими нейронными сетями.

Библиотеки TensorFlow и Keras – используются для построения и обучения нейронных сетей. Позволяют создавать сложные модели для распознавания

лиц, использовать предварительно обученные модели и настраивать их под конкретные задачи.

Библиотека PyTorch – используется для глубинного обучения, аналогичная TensorFlow, с удобным интерфейсом и гибкостью. Поддерживает сложные архитектуры нейронных сетей, интегрируется с другими инструментами для обработки изображений.

Библиотека Face_recognition – разработана на основе dlib, простая в использовании для распознавания лиц. Предоставляет функции для обнаружения, распознавания и сравнения лиц.

1.2 Анализ существующих решений

В настоящее время существует множество аналогов разрабатываемого приложения, однако каждое из них обладает лишь ограниченным набором функций. Проведем анализ существующих приложений и создадим сравнительную таблицу для выявления основных характеристик для создания приложения для магистерской диссертации.

1.2.1 Биометрия Windows

Система биометрической аутентификации Windows-Hello, созданная компанией Microsoft для операционных систем Windows 10 и Windows 11. Она позволяет пользователям безопасно и удобно входить в систему без ввода паролей, используя следующие биометрические данные:

- Распознавание лица – применяет инфракрасные камеры для сканирования и идентификации лица пользователя.

- Отпечаток пальца – использует встроенные или внешние сканеры отпечатков пальцев.

- Радужная оболочка глаза – в некоторых устройствах поддерживается сканирование радужной оболочки глаза.

Система обеспечивает быстрый и защищенный доступ к устройствам, приложениям и данным, используя технологии, которые значительно сложнее подделать по сравнению с обычными паролями. Это повышает уровень безопасности и удобства для пользователей.

Плюсы решения:

- Безопасность. Windows Hello предлагает биометрическую аутентификацию, такую как сканирование лица или отпечатков пальцев, что обеспечивает более высокий уровень безопасности, чем традиционные пароли.

- Удобство. Пользователи могут легко и быстро войти в систему, используя свои биометрические данные, без необходимости запоминания или ввода паролей.

- Интеграция. Windows Hello интегрирована в операционную систему Windows, что позволяет использовать ее для различных задач, таких как вход в учетную запись Microsoft, вход в приложения и т. д.

Минусы решения:

- Оборудование. Для использования Windows Hello требуется специальное оборудование, такое как камера с функцией распознавания лица или сканер отпечатков пальцев. Это может быть дополнительной стоимостью для пользователей.

- Приватность. Некоторые пользователи могут опасаться использования биометрических данных из-за потенциальных проблем с приватностью и безопасностью.

1.2.2 Динамическая блокировка Windows (Dynamic Lock)

Функция безопасности, доступная в Windows 10 и Windows 11, которая автоматически блокирует компьютер, когда пользователь покидает его. Принцип работы блокировки, следующий:

- Сопряжение через Bluetooth – функция устанавливает Bluetooth-соединение между компьютером и мобильным устройством пользователя (например, смартфоном или смарт-часами).

- Детекция удаления – когда сопряженное устройство выходит из зоны действия Bluetooth (обычно несколько метров), Windows автоматически блокирует компьютер через минуту, предотвращая несанкционированный доступ.

- Динамическая блокировка помогает защитить компьютер от несанкционированного использования, когда пользователь отсутствует, предоставляя до-

полнительный уровень безопасности без необходимости вручную блокировать систему.

Плюсы решения:

- Безопасность. Динамическая блокировка Windows автоматически блокирует компьютер, когда устройство пользователя (например, смартфон) находится вне зоны действия Bluetooth, что помогает защитить данные от несанкционированного доступа.

- Удобство. Это предоставляет дополнительный уровень безопасности без необходимости вводить пароли или блокировать компьютер вручную, когда пользователь уходит от компьютера.

Минусы решения:

- Требования к оборудованию. Для использования динамической блокировки Windows необходимо, чтобы компьютер и устройство пользователя поддерживали технологию Bluetooth, что может быть проблемой для некоторых пользователей или организаций.

- Недостаточная надежность. В некоторых случаях динамическая блокировка может не работать надежно из-за проблем с подключением Bluetooth или других технических проблем, что может вызывать неудобства для пользователей.

1.2.3 Приложение KeyLemon

Это программное обеспечение для настольных компьютеров, которое позволяет разблокировать Windows с помощью камеры. Оно отличается высокой точностью и безопасностью, предотвращая попытки обмана системы с использованием фотографий пользователя. Основные характеристики KeyLemon включают:

- Идентификация лица – программа использует камеру для распознавания лица пользователя.

- Защита от подделок – возможность обнаружения попыток обмана с помощью фотографий.

- Дополнительные проверки безопасности – возможность добавления сложных действий, таких как моргание или движение головы, для повышения уровня безопасности.

- Шифрование биометрических данных – биометрическая информация сохраняется и шифруется на локальном устройстве, что обеспечивает защиту данных.

Несмотря на то, что KeyLemon больше не поддерживается, его всё ещё можно скачать и использовать с определёнными ограничениями.

Плюсы решения:

- Безопасность. KeyLemon предлагает удобную и безопасную альтернативу традиционным паролям. Распознавание лица обеспечивает более высокий уровень безопасности, так как сложнее подделать или украсть биометрические данные, чем пароль.

- Удобство. Пользователи могут быстро и легко войти в систему, просто показав свое лицо перед веб-камерой, без необходимости запоминания или ввода пароля.

- Интеграция. KeyLemon интегрируется в операционную систему и позволяет использовать биометрическую аутентификацию для входа в систему Windows или Mac, а также для доступа к различным приложениям и сайтам.

- Многофункциональность. KeyLemon может также предоставлять дополнительные функции, такие как отслеживание активности пользователя или автоматическое блокирование системы при отсутствии пользователя.

Минусы решения:

- Точность. В зависимости от условий освещенности и качества веб-камеры, точность распознавания лица может быть ниже, чем у других биометрических методов, таких как сканеры отпечатков пальцев.

- Приватность. Некоторые пользователи могут быть обеспокоены использованием распознавания лица из-за потенциальных проблем с приватностью и безопасностью, так как их биометрические данные могут быть скомпрометированы или использованы без их согласия.

- Требования к оборудованию. Для использования KeyLemon требуется наличие веб-камеры на компьютере или ноутбуке, что может быть проблемой для пользователей, у которых такого оборудования нет или оно не работает должным образом.

- Самый главный минус – отсутствие обновлений – данное приложение больше не поддерживается разработчиком, поэтому новых версий этого приложения больше не будет. Помимо версий, возникнут проблемы и с безопасностью – методы защиты данных также не подлежит обновлению.

1.2.4 Приложение True Key

Это программное обеспечение для управления паролями, разработанное компанией McAfee. Оно позволяет пользователям хранить свои пароли и другие данные аутентификации в безопасном цифровом хранилище и автоматически заполнять их при входе на веб-сайты и в приложения. Одним из ключевых преимуществ True Key является множественная аутентификация, позволяющая использовать различные методы аутентификации, такие как сканирование лица, отпечатка пальца, распознавание голоса и другие, помимо обычного ввода пароля. Это увеличивает уровень безопасности аккаунта. True Key также синхронизирует данные между устройствами пользователя, обеспечивая доступ к паролям с любого устройства.

Плюсы решения:

- Удобство использования. True Key предлагает удобный способ управления паролями. Он хранит все ваши пароли в безопасном цифровом хранилище и автоматически заполняет их при входе на веб-сайты и в приложения.

- Множественная аутентификация. Помимо пароля, True Key предлагает различные методы аутентификации, такие как сканирование лица, отпечатка пальца, распознавание голоса и другие. Это повышает безопасность вашего аккаунта.

- Синхронизация между устройствами. True Key синхронизирует ваши пароли и данные аутентификации между устройствами, что позволяет получить к ним доступ с любого устройства.

- Генератор безопасных паролей: Приложение предлагает генератор безопасных паролей, который помогает создавать уникальные и сложные пароли для каждого аккаунта.

Минусы решения:

- Стоимость. Хотя у True Key есть бесплатная версия, некоторые расширенные функции могут быть доступны только в платной версии. Это может быть дополнительным расходом для пользователей, а также негативным опытом, так как, не всегда бесплатная версия работает хорошо без расширений.

- Приватность и безопасность. Поскольку True Key хранит все ваши пароли в облаке, есть потенциальный риск компрометации данных, если ваш аккаунт будет скомпрометирован или возникнут проблемы с безопасностью в облачном хранилище.

- Зависимость от подключения к интернету. Для использования True Key требуется подключение к интернету. Если вы оказались в месте без доступа к всемирной сети, то можете столкнуться с проблемами при доступе к вашим паролям.

- Интеграция с некоторыми приложениями. Некоторые пользователи могут столкнуться с проблемами интеграции True Key с некоторыми приложениями или веб-сайтами, что может снизить удобство использования.

1.2.5 Приложение Rohos Face Logon

Это программное обеспечение для биометрической аутентификации, которое позволяет пользователям входить в свою систему Windows, используя распознавание лица. Оно предлагает альтернативу вводу пароля при входе в систему и обеспечивает более удобный и безопасный способ доступа к компьютеру.

Принцип работы Rohos Face Logon основан на анализе геометрии лица пользователя. После установки программы пользователь регистрирует свое лицо, а затем при каждой попытке входа в систему программа сканирует и сравнивает лицо с зарегистрированным образцом для аутентификации. При несовпадении система дальше вас не пропустит.

Основные особенности Rohos Face Logon включают в себя:

- Безопасность. Программа предлагает более надежный метод аутентификации по сравнению с традиционным вводом пароля, так как сложнее подделать или украсть биометрические данные пользователя.

- Удобство. Вход в систему осуществляется автоматически при распознавании лица, что делает процесс более быстрым и удобным, а также более безопасным.

- Многофункциональность. Помимо входа в систему, Rohos Face Logon может использоваться для блокировки и разблокировки компьютера во время его использования.

- Легкость использования. Программа имеет простой и интуитивно понятный интерфейс, что делает ее доступной для широкого круга пользователей.

Минусы решения:

- Надежность распознавания лица. Точность распознавания лица может быть ниже, чем у других биометрических методов, особенно при изменении условий освещенности или угла обзора веб-камеры. Это может привести к тому, что пользователи будут испытывать проблемы с входом в систему или необходимостью повторной аутентификации.

- Зависимость от оборудования. Работа Rohos Face Logon полностью зависит от наличия камеры веб-камеры на компьютере или ноутбуке. Если камера не работает или отсутствует, это может сделать программу непригодной для использования.

- Приватность. Некоторые пользователи могут быть обеспокоены использованием биометрических данных из-за потенциальных проблем с приватностью и безопасностью. Если данные лица скомпрометированы, это может привести к утечке персональной информации.

- Необходимость дополнительного времени на настройку. Для эффективной работы программы требуется некоторое время на настройку и регистрацию. Некоторым пользователям может показаться, что это занимает слишком много времени или представляет собой неудобство.

- Сложности в многопользовательской среде. Если на компьютере несколько пользователей, каждому из них придется настраивать Rohos Face Logon отдельно. Это может вызвать сложности в управлении доступом и обслуживании системы.

1.2.6 Приложение Luxand’s Blink!

Это программное обеспечение для биометрической аутентификации, разработанное компанией Luxand. Оно позволяет пользователям входить в свою систему Windows, используя распознавание лица.

Принцип работы Blink! основан на сканировании и анализе геометрии лица пользователя. После установки программы пользователь регистрирует свое лицо, а затем при каждой попытке входа в систему программа сканирует и сравнивает лицо с зарегистрированным образцом для аутентификации.

Основные особенности Luxand's Blink! включают в себя:

- Безопасность. Приложение предлагает более надежный метод аутентификации по сравнению с традиционным вводом пароля, так как сложнее подделать или украсть биометрические данные пользователя.

- Удобство. Вход в систему осуществляется автоматически при распознавании лица, что делает процесс более удобным и быстрым.

- Многофункциональность. Помимо входа в систему, Luxand's Blink! может использоваться для блокировки и разблокировки компьютера во время его использования.

- Легкость использования. Программа имеет простой и интуитивно понятный интерфейс, что делает ее доступной для широкого круга пользователей.

Недостатки решения:

- Надежность распознавания лица. Точность распознавания лица может быть снижена в зависимости от условий освещенности, угла обзора камеры и других факторов. Это может вызвать проблемы с входом пользователя в систему или требовать повторной аутентификации.

- Зависимость от оборудования. Работоспособность Luxand's Blink! полностью зависит от наличия веб-камеры на компьютере или ноутбуке. Отсут-

ствие или неисправность камеры может сделать программу неприменимой для использования.

- Приватность и безопасность. Некоторые пользователи могут быть обеспокоены использованием биометрических данных из-за потенциальных проблем с приватностью и безопасностью. В случае утечки данных лица пользователь может столкнуться с потенциальной утечкой личной информации.

- Требование дополнительного времени на настройку. Для эффективной работы программы может потребоваться дополнительное время на настройку и регистрацию лица пользователя. Это может быть воспринято некоторыми пользователями как неудобство или слишком трудоемкий процесс.

- Сложности в многопользовательской среде. Если на компьютере несколько пользователей, каждый из них должен будет настраивать Luxand's Blink! индивидуально. Это может вызвать сложности в управлении доступом и обслуживании системы.

1.3 Сравнительный анализ

Проанализируем существующие решения для лучшего понимания различий между приложениями, а также выявления нужных функций для разработки.

Критерии для анализа выбраны следующие:

- Тип аутентификации – способ, которым приложение идентифицирует пользователя (например, по лицу, отпечатку пальца, радужке глаза и т.д.).

- Поддерживаемые устройства – операционные системы и устройства, на которых приложение может быть установлено и использовано.

- Безопасность – общий уровень безопасности, который обеспечивает приложение, включая использование аппаратных и программных методов защиты.

- Легкость использования – насколько просто и удобно приложение для конечного пользователя.

- Стоимость – стоимость использования приложения, включая бесплатные и платные версии. Возможен вариант бесплатной демоверсии приложения с ограниченным функционалом.

- Дополнительные функции – наличие дополнительных функций, которые предлагает приложение, помимо основной аутентификации. Например, менеджер паролей.

- Надежность – общая стабильность и надежность работы приложения.

- Скорость аутентификации – время, необходимое для идентификации пользователя с помощью приложения.

- Простота настройки – уровень сложности процесса установки и настройки приложения для использования. Очень часто из-за сложности установки приложения становятся невостребованными, так как не все пользователи готовы тратить время на чтение инструкций или смотреть обучающие видео, а неправильная установка или настройка может привести к некорректной работе приложения.

- Возможность однократной идентификации – данная функция позволяет идентифицировать пользователя в момент первого входа в систему, а также в момент выхода устройства из спящего режима. Обычно, после прохождения однократной идентификации, приложение более никак себя не проявляет, если в нём нет каких-то дополнительных функций. Также, однократная идентификация (при её успешном прохождении) не может, в дальнейшем, помешать нарушителю пользоваться компьютером.

- Возможность постоянной идентификации – функция, при которой используется не однократная идентификация, а постоянная. То есть, пока камера устройства фиксирует лицо пользователя в видеопотоке, программа его распознаёт через заданные промежутки времени, а также прекращает анализ видеопотока и сам видеопоток, после потери лица (пользователь покинул зону действия камеры).

- Сопряжение с телефоном – возможность использования распознавателя дистанционно с помощью мобильного устройства (например, используя стороннее приложение на смартфоне или используя bluetooth-сопряжение).

Результат анализа представлен в качестве сравнительной таблицы (таблица 1).

Таблица 1 – Сравнительный анализ существующих приложений

Характеристика	Windows Hello	Динамическая блокировка Windows	KeyLemon	True Key	Rohos Face Logon	Luxand's Blink!
Тип аутентификации	Биометрическая (лицо, отпечаток пальца, радужка глаза)	Bluetooth-сопряжение	Лицо	Лицо, пароль, PIN, отпечаток пальца	Лицо	Лицо
Поддерживаемые устройства	Windows 10 и выше	Windows 10 и выше	Windows, macOS	Windows, macOS, iOS, Android	Windows	Windows
Безопасность	Высокая, благодаря интеграции с аппаратным обеспечением	Средняя, зависит от сопряженного устройства	Средняя	Высокая, благодаря многослойной аутентификации	Средняя	Средняя
Легкость использования	Очень высокая	Высокая	Средняя	Высокая	Средняя	Средняя
Стоимость	Бесплатно	Бесплатно	Платно	Бесплатно, платные функции	Платно	Платно
Дополнительные функции	Интеграция с Microsoft Account, поддержка бизнес-учеток	Автоматическая блокировка при удалении сопряженного устройства	Поддержка мульти-учеток	Поддержка многофакторной аутентификации, синхронизация устройств	Поддержка смарт-карт	Поддержка нескольких учеток
Надежность	Очень высокая	Высокая	Средняя	Высокая	Средняя	Средняя
Скорость аутентификации	Быстрая	Быстрая	Средняя	Быстрая	Средняя	Средняя
Простота настройки	Простая	Простая	Средняя	Простая	Средняя	Средняя
Однократная идентификация (после входа)	Да	Нет	Да	Да	Да	Да
Постоянная идентификация	Нет	Да	Нет	Нет	Нет	Нет
Сопряжение с телефоном	Нет	Да	Нет	Да	Нет	Нет

1.4 Постановка задачи

На основе проведённого анализа существующих решений составлен список основных функций, которые должны быть у пользовательского решения:

- Распознавание лица пользователя во время его работы за компьютером.

- База данных, которая хранит фото и идентификаторы на локальном и (или) на удалённом сервере.

- Приложение работает без поддержки интернета (возможна потеря некоторого функционала при отключении от всемирной сети).

- Блокировка доступа незарегистрированным пользователям во время отсутствия зарегистрированного пользователя (пользователей) у экрана.

- Реализация однопользовательского и многопользовательского режимов работы приложения.

- Фиксация действий пользователей – аналог аудита, позволяющего узнать, какие приложения использовались.

- Возможность удалённого доступа к базе данных.

- Возможность удалённого получения снимка или записи с экрана персонального компьютера или ноутбука.

- Управление базой данных – её редактирование и удаление записей.

2 АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ С ИСПОЛЬЗОВАНИЕМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

2.1 Предлагаемый алгоритм решения задачи

Самым главным алгоритмом в программе является распознавание лица пользователя. Если программа распознает пользователя, значит, доступ будет разрешен, в противном случае, программа заблокирует доступ постороннему.

Представим алгоритм в виде нескольких шагов.

Шаг первый – захват изображения с веб-камеры при наличии в ракурсе человека. Программа захватывает изображение с веб-камеры, которая находится в устройстве пользователя, или это может быть внешняя веб-камера, для его дальнейшего анализа.

Шаг второй – предобработка изображения. Полученное изображение может быть подвергнуто предварительной обработке для улучшения качества. Многие средства распознавания могут изменять яркость и контрастность изображения, а также, достроить 2д изображение до 3д картинки. Также, программа определит на фотографии область с лицом пользователя – это будет использовано в следующем шаге.

Шаг третий – сравнение с зарегистрированными пользователями. Полученное изображение сравнивается с изображениями пользователей из базы данных.

Шаг четвёртый – определение совпадения. Если найдено совпадение между изображением с веб-камеры и изображениями пользователей из базы данных, то пользователь идентифицируется как один из пользователей системы.

Шаг пятый – принятие решения. На основе результатов сравнения принимается решение о дальнейших действиях. Если пользователь идентифицирован, ему может быть предоставлен доступ к системе, если пользователь не идентифицирован, программа может предложить повторить попытку или отклонить доступ.

2.2 Обоснование выбора программно-технического обеспечения

2.2.1 Язык программирования

Для создания приложения, которое распознает лица и блокирует доступ не зарегистрированным пользователям, можно использовать различные языки программирования. Вот несколько из них:

- Python – популярный язык программирования с обширной поддержкой библиотек для обработки изображений и машинного обучения, таких как OpenCV и TensorFlow. С помощью этих библиотек можно реализовать алгоритмы распознавания лиц и разработать приложение для блокировки доступа на основе распознавания лиц.

Преимущества языка Python:

- Простота в использовании и чистый синтаксис.
- Обширная поддержка библиотек для обработки изображений и машинного обучения, таких как OpenCV и TensorFlow.
- Подходит как для прототипирования, так и для разработки более сложных систем.

Недостаток языка Python – не самый быстрый язык из рассмотренных, что может быть проблематично для приложений, требующих высокой производительности.

- C++ является еще одним популярным языком, который может быть использован для создания приложения распознавания лиц. Он предоставляет высокую производительность и позволяет написать оптимизированный код. Библиотеки, такие как OpenCV, имеют свои версии для C++, что делает его хорошим выбором для таких приложений.

Преимущества языка C++:

- Очень высокая производительность благодаря низкоуровневому доступу к ресурсам компьютера.
- Широкая поддержка библиотек для компьютерного зрения, таких как OpenCV.
- Подходит для разработки высокопроизводительных приложений.

Недостатки языка C++:

- Сложнее в изучении и разработке, чем Python.
- Требует более длительного времени разработки.

- Java – мощный язык программирования, который широко используется для создания приложений с использованием машинного обучения и компьютерного зрения. Множество библиотек, таких как OpenCV и Deeplearning4j, имеют свои версии для Java, что делает его подходящим для создания приложения распознавания лиц.

Преимущества языка Java:

- Портативность и широкая поддержка для различных платформ.
- Большое количество библиотек для обработки изображений и машинного обучения.
- Подходит для создания кроссплатформенных приложений.

Недостатки языка Java:

- Менее производителен по сравнению с некоторыми другими языками, такими как C++.
- Меньшее количество инструментов и библиотек по сравнению с Python.

Исходя из основных преимуществ и недостатков был выбран язык Python, так как при разработке важны:

- Простота и удобство использования. Python известен своей простотой и лаконичностью, что делает его отличным выбором для начинающих разработчиков. Благодаря чистому и интуитивно понятному синтаксису, программирование на Python становится проще и приятнее.

- Большое количество библиотек и фреймворков. Python обладает богатой экосистемой библиотек и фреймворков для разработки приложений по распознаванию лиц, таких как OpenCV, dlib, face_recognition и многие другие. Эти библиотеки обеспечивают широкие возможности для работы с изображениями, обработки видео и реализации алгоритмов распознавания лиц. Также, есть библиотеки, которые позволяют написать чат-бота.

- Скорость разработки. Python позволяет разрабатывать ПО со сложной структурой и большим количеством функций в кратчайшие сроки, не затрачивая большое количество ресурсов.

- Поддержка машинного обучения. Python широко используется в области машинного обучения и искусственного интеллекта благодаря популярным библиотекам, таким как TensorFlow, Keras, PyTorch и Scikit-learn. Эти библиотеки предоставляют мощные инструменты для обучения моделей распознавания лиц и применения их в приложениях.

- Поддержка различных платформ. Python поддерживается на всех основных операционных системах (Windows, macOS, Linux) и может быть использован для создания кроссплатформенных приложений по распознаванию лиц.

2.2.2 Используемые библиотеки

Так как для создания приложения был выбран язык Python, то для разработки понадобятся некоторые библиотеки, которые поддерживают работу с этим языком:

- Tkinter – стандартный пакет для создания графического пользовательского интерфейса (GUI) в Python. Он предоставляет простой способ создания окон, кнопок, меню, рамок и других элементов интерфейса для ваших приложений. Tkinter основан на библиотеке Tcl/Tk, которая широко используется для создания GUI во многих других языках программирования. Tkinter обеспечивает переносимый и легко настраиваемый способ создания пользовательских интерфейсов на Python.

- Python Imaging Library (PIL) – это библиотека Python для работы с изображениями. Она позволяет открывать, обрабатывать и сохранять изображения в различных форматах. PIL предоставляет множество функций для редактирования изображений, таких как изменение размера, поворот, наложение фильтров, обрезка и многое другое. Благодаря PIL вы можете проводить обработку изображений в ваших Python-приложениях.

- Face_recognition – это библиотека для распознавания лиц в изображениях с помощью Python и dlib. Она предоставляет простой способ распознавания

и идентификации лиц на фотографиях. Face_recognition использует нейронную сеть для выявления и сопоставления лиц на изображениях. Она позволяет определять расположение лиц, их признаки и идентифицировать конкретные лица на фотографиях.

- OpenCV (Open Source Computer Vision Library) – это библиотека компьютерного зрения с открытым исходным кодом, предназначенная для обработки изображений и видео. OpenCV предоставляет богатые возможности для работы с изображениями, такие как детекция объектов, распознавание лиц, трекинг объектов, определение образцов, робототехника и многое другое. Она поддерживает различные языки программирования, включая языки семейства C и Python, и может быть использована для создания разнообразных приложений компьютерного зрения.

- AIogram – это библиотека для Python, которая предоставляет инструменты для создания ботов Telegram с использованием асинхронного программирования. Эта библиотека облегчает создание и управление ботами, обрабатывает взаимодействие с пользователем, отправку сообщений и другие функции, связанные с Telegram API.

2.2.3 Среда разработки PyCharm IDE

Так как был выбран язык программирования Python, то и платформу для разработки следует выбирать именно под него.

PyCharm – мощная интегрированная среда разработки, специально разработанная для удобной работы с языком программирования Python. При создании программы по распознаванию лиц, PyCharm предоставляет ряд преимуществ и возможностей.

Преимущества PyCharm. Во-первых, PyCharm обладает продвинутой интеграцией с различными библиотеками и фреймворками, которые могут быть использованы для создания программы по распознаванию лиц. Например, PyCharm легко интегрируется с библиотекой OpenCV, которая является одной из самых популярных библиотек для работы с изображениями и видео, включая задачи компьютерного зрения, такие как распознавание лиц.

Кроме того, PyCharm обеспечивает удобную работу с отладчиком, автодополнением кода, инструментами анализа кода, автоматическим форматированием и другими возможностями, которые значительно упрощают процесс разработки. Это позволяет быстро и эффективно создавать и тестировать программу по распознаванию лиц без лишних сложностей.

Кроме того, PyCharm имеет широкий выбор плагинов, которые могут быть установлены для расширения функциональности среды разработки. Это дает возможность использовать специализированные инструменты и ресурсы, которые могут значительно улучшить процесс разработки программы по распознаванию лиц.

Таким образом, PyCharm обладает всем необходимым для создания программа по распознаванию лиц, предоставляя разработчику удобное и эффективное рабочее окружение, интегрированные инструменты и возможности для ускорения процесса разработки и повышения качества конечного продукта.

2.2.4 База данных Mongo

Уже при проектировании приложения, для реализации некоторых функций, выяснилась потребность в создании базы данных, однако, обычная, реляционная, база данных не подойдёт по ряду причин. Одна из них – наличие не структурированных данных, которые реляционными базами данных обрабатываться не могут. Также, скорость обработки данных в реляционных базах не является высокой, что может тормозить работу программы уже на стадии распознавания идентификатора пользователя. Свою роль играет и скорость разработки – при создании базы данных структура может кардинально поменяться несколько раз.

Оптимальным решением будет использование не реляционной базы данных, которая имеет ряд преимуществ:

- подходит для хранения больших объемов неструктурированной информации, а также хороша для быстрой разработки и тестирования гипотез;

- может хранить данные любого типа и добавлять новые в процессе работы;

- имеет распределенную архитектуру, поэтому хорошо масштабируется горизонтально и отличается высокой производительностью.

MongoDB предоставляет отличное решение для хранения и управления большим объемом данных, что делает его идеальным выбором для программ по распознаванию лиц. В отличие от реляционных баз данных, MongoDB использует гибкую структуру документов в формате JSON, что позволяет эффективно хранить и обрабатывать структурированные и неструктурированные данные, такие как изображения лиц и их метаданные.

Одним из ключевых преимуществ MongoDB является его горизонтальное масштабирование, которое позволяет легко увеличивать емкость базы данных при необходимости, обеспечивая высокую производительность и доступность данных. Это особенно важно для программ по распознаванию лиц, которые могут обрабатывать огромные объемы информации.

MongoDB также обладает мощными инструментами для запросов и агрегации данных, что делает его идеальным для анализа больших объемов данных и извлечения ценной информации из них. Благодаря возможности использования различных индексов и запросов, MongoDB обеспечивает быстрый доступ к данным, что особенно важно для приложений по распознаванию лиц.

В обычных реляционных базах данных информация хранится в виде взаимосвязанных таблиц. Их структура жёстко задана, и поменять её непросто. Строки каждой таблицы имеют одинаковый набор полей, данные обрабатывают с помощью запросов на языке SQL.

Эти базы наглядны, но не всегда удобны – например, в тех случаях, когда вам нужно хранить информацию без определённой структуры: представить её в виде двумерных таблиц нельзя.

В MongoDB всё устроено немного по-другому. Базы состоят из коллекций и документов – иерархических структур, содержащих пары «ключ – значение» (поля).

Если проводить аналогии с реляционной базой, коллекции при таком способе хранения соответствуют таблицам, а документы – строкам.

У документов нет строгой структуры. Они могут содержать разные наборы полей, причём различающиеся как по типу, так и по количеству.

В целом, MongoDB представляет собой мощное и гибкое решение для создания программ по распознаванию лиц, обеспечивая высокую производительность, масштабируемость и отказоустойчивость базы данных.

2.3 Модель жизненного цикла

Современные методологии разработки программного обеспечения, несмотря на свои преимущества и недостатки, зачастую не соответствуют требованиям конкретных задач. В связи с этим, набирают популярность гибкие методологии, такие как Agile.

Основными принципами Agile являются:

- Удовлетворенность заказчика – достигается за счет быстрых релизов и постоянного взаимодействия с бизнесом.

- Открытость к изменениям – проект легко адаптируется к новым требованиям и условиям.

- Частые релизы – новая версия продукта выпускается каждый период, длящийся от двух до восьми недель.

- Ежедневное общение – разработчики тесно взаимодействуют с заказчиком и бизнесом.

- Мотивация разработчиков – достигается за счет самоуправления и отсутствия микроменеджмента.

- Прямая коммуникация – лучший способ решения проблем и повышения эффективности.

- Работающий продукт – главный показатель прогресса.

- Постоянный прогресс – проект непрерывно совершенствуется.

- Высокое качество проектирования – повышает гибкость и адаптивность проекта.

- Минимизация лишней работы – фокус на результатах и эффективности.

- Автономные команды – самоорганизуются и принимают решения самостоятельно.

- Постоянный анализ – непрерывное улучшение качества работы.

Преимущества Agile:

- Гибкость и адаптивность – быстрая реакция на изменения и новые требования.

- Сниженные риски провала – частые релизы и тесное общение с заказчиком позволяют своевременно корректировать проект.

- Устойчивость к срыву сроков – гибкое планирование позволяет адаптировать сроки к реальному ходу разработки.

- Высокая вовлеченность команды – самоуправление и тесная работа с бизнесом повышают мотивацию и эффективность.

- Быстрая реакция на проблемы – баги и ошибки устраняются в следующем цикле разработки.

- Минимизация рутины – фокус на работе над проектом, а не на документации и отчетах.

Недостатки Agile

- Отсутствие четкого плана – конечный результат может отличаться от первоначального плана.

- Потребность в тесном общении – заказчик должен активно участвовать в проекте.

- Зависимость от команды – смена разработчиков или руководителя может затруднить продолжение работы.

- Фокус на деталях – иногда можно упустить из виду глобальную цель проекта.

- Сложности с внедрением – переход от другой методологии может потребовать времени и усилий.

Несмотря на некоторые недостатки, Agile является современной и эффективной методологией разработки программного обеспечения. В рамках магистерской диссертации, где гибкость и адаптивность являются важными факторами, а общаться с заказчиком нет необходимости. Agile представляется наиболее подходящим выбором.

2.4 Используемая архитектура

Исходя из постановки задачи, можно сделать вывод, что разработка предполагает наличие двух программных продуктов, а именно программы на персональный компьютер и Telegram-бота, которые должны быть связаны между собой при помощи базы данных. Для этого нужно выбрать подходящую архитектуру.

Микросервисная архитектура предполагает разработку и поддержку приложений с использованием небольших модульных сервисов, а не создание программного обеспечения в виде одного большого унифицированного блока кода (монолита). Основная концепция архитектуры в том, чтобы разделить сложное приложение на несколько небольших автономных и управляемых компонентов. Это позволяет повысить гибкость разработки, улучшить отказоустойчивость и облегчить поддержку приложения.

Каждый микросервис имеет собственный набор кода, базу данных и API для взаимодействия с другими сервисами. Они могут быть написаны на разных языках программирования и использовать различные технологии. Взаимодействие сервисов может осуществляться посредством сетевых запросов или сообщений.

Применение микросервисной архитектуры позволяет децентрализовать организацию команд разработчиков. Разные команды могут взаимодействовать с отдельными сервисами и не затрагивать работу соседей. Это способствует ускорению разработки и позволяет сосредоточиться на специфических потребностях проекта.

Ключевые особенности микросервисной архитектуры:

- Масштабируемость. Микросервисы могут быть масштабированы независимо друг от друга. Это позволяет распределять нагрузку и ресурсы по сервисам, которые нуждаются в поддержке прямо сейчас.

- Независимость. Каждый сервис полностью автономен и не затрагивает работу соседей. Это означает, что каждый сервис может быть разработан, развернут и обновлен отдельно, без воздействия на остальные компоненты.

- Легковесность. Микросервисы используют легковесные протоколы для взаимодействия между собой, такие как REST или gRPC. Это позволяет сервисам быстро обмениваться данными.

- Гибкость. Микросервисы могут быть разработаны с использованием разных технологий и языков программирования. Это дает разработчикам большую гибкость при выборе технологий для конкретных компонентов системы. Это также позволяет эффективнее управлять командой разработки.

- Управление ошибками. У каждого сервиса есть свое собственное управление ошибками и восстановлением после сбоев. Если один сервис не отвечает, это не приводит к полной остановке системы. Например, сайт продолжит принимать покупки, если микросервис, отвечающий за логистику, будет какое-то время недоступен.

- Легкость развертывания. Микросервисы могут быть легко развернуты на различных серверах или облачных платформах.

- Распределенная разработка. При построении микросервисной архитектуры разработка приложения может быть распределена между несколькими командами. Каждая команда может работать над отдельным сервисом, что ускоряет разработку.

- Легкая замена. Если требуется заменить один сервис, его можно легко заменить, не затрагивая другие сервисы. Это упрощает обновление системы и добавление новых функций.

Несмотря на многочисленные преимущества, микросервисная архитектура также имеет ряд особенностей, которые затрудняют интеграцию такого подхода.

Несмотря на то, что микросервисы – более современный подход к построению архитектуры – это не значит, что он должен внедряться везде, вне зависимости от бизнеса и масштабов приложения. Правильнее говорить о спецификации и стратегических задачах проекта. Решившись распилить монолит, разработчики должны четко понимать, зачем они ввязываются в этот сложнейший процесс.

Рассмотрим сильные и слабые стороны подходов (Таблица 2). Оба подхода имеют свои сильные и слабые стороны, и выбор между ними зависит от конкретного проекта и его требований.

Таблица 2 – Отличия архитектур

Параметры	Монолитная архитектура	Микросервисная архитектура
Размер и сложность	Одно большое приложение, в котором все компоненты связаны друг с другом напрямую	Приложение разбивается на множество микросервисов, каждый из которых отвечает за конкретную функциональность. Это позволяет разделить сложное приложение на более простые и независимые части
Гибкость и масштабируемость	Изменения и масштабирование требуют работы со всем приложением, что может быть неэффективным и сложным	Микросервисная архитектура дает большую гибкость при разработке и поддержке приложения. Каждый микросервис может быть разработан, протестирован и развернут независимо от других. Это также облегчает масштабирование, поскольку можно масштабировать только необходимые компоненты
Надежность	Если одна часть приложения отказывает, это может привести к отказу всего приложения	Если один микросервис перестает работать, остальные микросервисы продолжают функционировать независимо
Сложность развертывания и управления	Монолитная архитектура может быть более простой в развертывании и управлении, поскольку ее элементы не связаны сложной системой зависимостей	Управление микросервисной архитектурой может быть сложнее, так как требуется управлять несколькими независимыми сервисами и их взаимодействием

Сложности работы с микросервисами:

- Управление. Микросервисная архитектура предполагает работу с большим количеством сервисов, каждый из которых имеет собственную версию и набор зависимостей.

- Комплексность взаимодействия. Микросервисы взаимодействуют друг с другом посредством сетевых запросов. Это может привести к проблемам с производительностью и надежностью, так как каждый сетевой запрос может стать точкой отказа.

- Обеспечение целостности данных. При использовании микросервисной архитектуры данные могут храниться и обрабатываться разными сервисами.

Обеспечение целостности данных и синхронизация между сервисами может быть сложной задачей.

- Сложность отладки и тестирования. В микросервисной архитектуре каждый сервис может быть разработан, развернут и масштабирован независимо. Это может затруднять процесс отладки и тестирования, так как необходимо изолировать проблему до конкретного сервиса.

- Уязвимости безопасности. Поскольку каждый сервис имеет доступ к части данных, уязвимость в одном сервисе может привести к компрометации всей системы.

Несмотря на минусы, данная архитектура подходит для разработки приложения.

2.5 Межсистемная интеграция

В магистерской диссертации рассматривается разработка приложения на персональном компьютере, а также разработка дополнительного инструмента, через который можно дистанционно использовать некоторые функции программы на компьютере. Однако, не все системы легко интегрируются друг в друга. Чтобы не было сложностей с синхронизацией используют буфер между приложениями, в котором идёт получение информации от приложений, их обработка и преобразование в нужный вид, а также отправка к нужному приложению.

API (Application Programming Interface) это программный интерфейс, благодаря которому одно приложение может взаимодействовать с другим. Эта технология сложна в эксплуатации, по сравнению с другими интерфейсами, но у неё есть существенное преимущество – скорость взаимодействия. Взаимодействие приложений не отражается в пользовательских интерфейсах, потому и выполняется значительно быстрее. Так, одна операция у API может занять сотую долю секунды, а в других значительно больше.

Шина API выполняет несколько ключевых функций:

- Маршрутизация запросов – шина API принимает запросы от клиентов и перенаправляет их к соответствующим сервисам или API.

- Аутентификация и авторизация - шина API может управлять процессами аутентификации и авторизации, проверяя права доступа клиентов к различным сервисам.

- Агрегация данных - она может объединять данные из различных сервисов в один ответ, упрощая клиентам получение нужной информации.

- Мониторинг и логирование - шина API обеспечивает мониторинг производительности и логирование всех запросов и ответов, что помогает в диагностике проблем и улучшении качества обслуживания.

- Кеширование – для повышения производительности шина API может кэшировать ответы на часто запрашиваемые данные.

- Политики управления трафиком – она может ограничивать количество запросов от одного клиента (rate limiting), чтобы предотвратить перегрузку системы.

Шина API играет ключевую роль в современной архитектуре микросервисов и распределенных систем, обеспечивая эффективное, безопасное и управляемое взаимодействие между различными компонентами системы. Она упрощает разработку, эксплуатацию и масштабирование API, что поможет при интеграции приложения, разрабатываемого в рамках магистерской диссертации.

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРЕДЛАГАЕМОГО АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ

Приложение – программа, предназначенная для выполнения определённых задач и рассчитанная на непосредственное взаимодействие с пользователем. Чаще всего приложения не могут работать с ресурсами компьютера напрямую, а делают это через операционную систему. В общем случае это программа, предназначенная для решения отдельных задач или класса задач, связанных с обработкой данных в определенной области деятельности.

Разработанное приложение является прикладным и предназначено для неподготовленного пользователя.

3.1 Функциональные требования

Функциональное требование – это заявление о том, как должна вести себя система. Он определяет, что система должна делать, чтобы удовлетворить потребности или ожидания пользователя. Функциональные требования можно рассматривать как функции, которые обнаруживает пользователь. Они отличаются от нефункциональных требований, которые определяют, как система должна работать внутри (например, производительность или безопасность).

Исходя из поставленной задачи, основными функциями будут считаться распознавание и блокировка нарушителей, а также работа с базой данных. Второстепенными функциями будут те, что дополняют основные, а также возможность использования Telegram-бота для манипуляций с базой данных и получения изображений с камеры. Так как есть разделение на основную программу (FaceID) и дополнительную в виде бота (Screenshotter), то функции каждой программы рассматриваются отдельно.

Функциональные требования для FaceID:

- Управление белым списком. Белый список – зарегистрированные пользователи, которые, как и администратор, могут пользоваться компьютером, однако, не имеют доступа к FaceID. Администратор может создавать, редактировать и удалять пользователей.

- Работа с файлообменником. Программа позволяет выбирать администратору изображение из памяти устройства для настройки экрана блокировки (это та самая картинка, которую увидит нарушитель при блокировке компьютера), также, можно выбрать фотографию для создания записи нового пользователя. Также, возможность сохранить изображение, полученное с веб-камеры.

- Настройка пароля для разблокировки компьютера и работы с ботом.

- Выбор места расположения поля для ввода пароля на экране компьютера.

- Настройка времени. Время задаётся дважды – первый раз, это установка временного промежутка, через который будет проводиться проверка пользователя. Второй раз время задаётся для установки времени ожидания перед первой проверкой (время будет отсчитываться с момента ввода пароля администратора).

- Работа с режимами. Пользователю предлагается несколько режимов работы, их можно комбинировать, либо выбрать только один. Первый режим – проверка всех пользователей – программа будет проверять всех пользователей, что зафиксирует камера (данный режим является основным, при его отключении могут быть задействованы другие режимы, что не связаны с распознаванием, например, «Шпион»). Второй режим является ограничением для первого – пользователь может настроить время общего доступа, то есть в установленный период времени компьютером сможет воспользоваться кто угодно. Третий режим – «Шпион», он включает запись всех действий, что совершались во время работы, в отдельный файл. Четвёртый режим – «Чёрный список» – это режим, который блокирует абсолютно всех пользователей, исключая администратора.

- Получение уведомления о вторжении. Если во время работы зарегистрированного пользователя в область действия камеры попадёт незарегистрированный пользователь, то система тут же уведомит об этом и заблокирует экран.

- Просмотр логов-отчётов. Программой ведётся запись действий пользователя в текстовый файл, который находится в папке с программой.

Функциональные требования для Screenshotter:

- Ввод идентификатора для доступа к просмотру главного меню функциональных возможностей бота.

- Ввод кода администратора для доступа к функциям бота.

- Удалённая работа с базой данных. Администратор может с помощью бота найти пользователя, создать нового пользователя, редактировать существующего пользователя, а также удалить данные о пользователе.

- Получение изображения с экрана компьютера. Администратор может запросить скриншот с экрана компьютера, либо видеозапись на указанное количество секунд.

3.1.1 Диаграмма прецедентов

UML (от англ. – Unified Modeling Language), как следует из названия – унифицированный язык моделирования для описания, визуализации и документирования объектно-ориентированных систем в процессе их анализа и проектирования. UML представляет собой набор соглашений, которые предназначены для облегчения процесса моделирования и обмена информацией в проектной группе.

Язык UML предоставляет стандартный способ написания проектной документации на системы, включая концептуальные аспекты, такие как бизнес-процессы и функции системы, а также конкретные аспекты, такие как выражения языков программирования, схемы баз данных и повторно используемые компоненты ПО.

Основу UML представляют диаграммы, которые различаются по типам и предназначены для моделирования различных аспектов разработки.

На основании выдвинутых функциональных требованиях выполним построение диаграммы прецедентов. Диаграмма прецедентов позволяет выполнить описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему. Кроме того, диаграмма прецедентов отображает множество прецедентов и действующих лиц, а также отношения между ними.

Прецедент – это не зависящее от реализации высокоуровневое представление того, что пользователь ожидает от системы, т.е. описание функциональности системы.

На диаграммах (рисунок 2 и рисунок 3) представлены функции, доступные администратору и пользователям (если имеются в белом списке), при использовании приложения на персональном компьютере и при использовании бота.

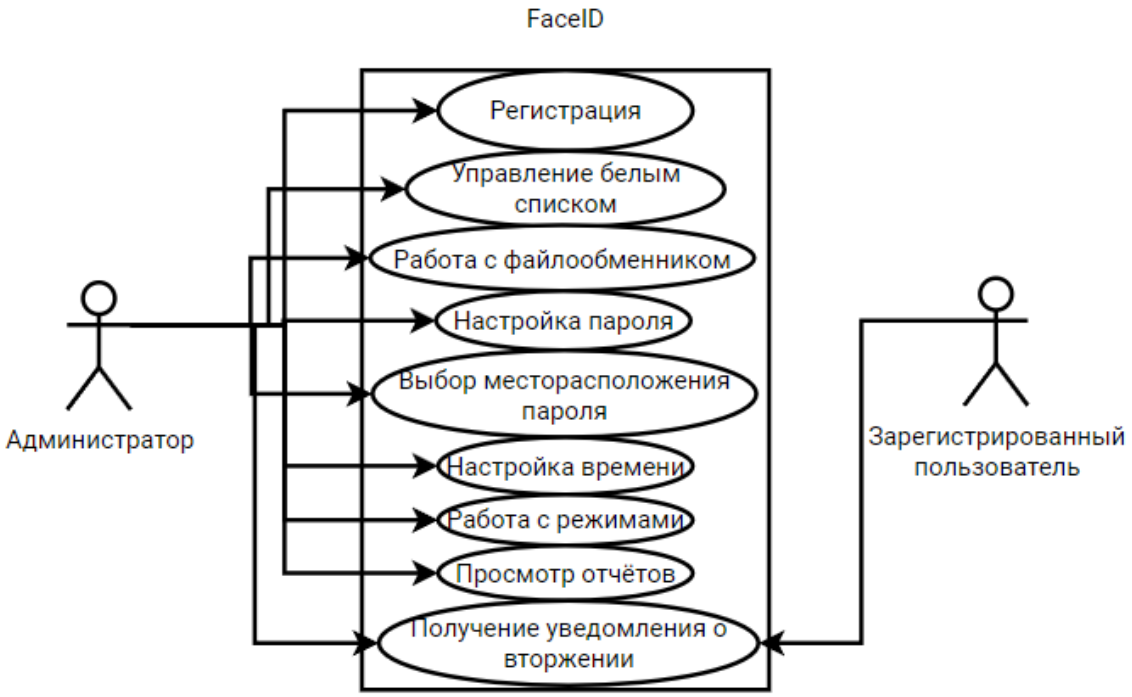

Рисунок 2 – Диаграмма прецедентов FaceID

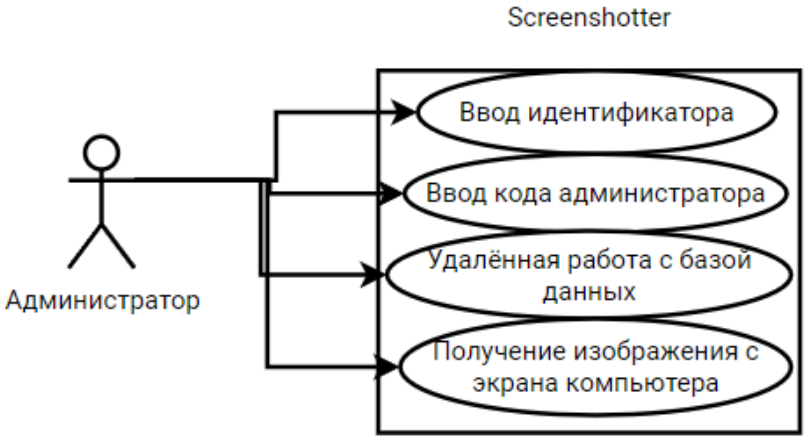

Рисунок 3 – Диаграмма прецедентов Screenshotter

3.1.2 Диаграмма последовательности

Диаграмма последовательности – такой тип диаграмм, в которой на единой временной оси отображается жизненный цикл объекта и поведение нескольких объектов информационной системы в рамках прецедента.

На данной диаграмме (рисунок 4) реализованы синхронные связи. То есть, администратор-отправитель передаёт ход управления программе-получателю, которой необходимо провести в прецеденте некоторое действие, а именно – настройка приложения. То есть, после установки программы, администратор её настраивает, переходя от одного окна к другому – настраивает экран блокировки, устанавливает пароль, устанавливает время проверки идентификатора, выбирает модули для работы приложения, а также настраивает «белый список».

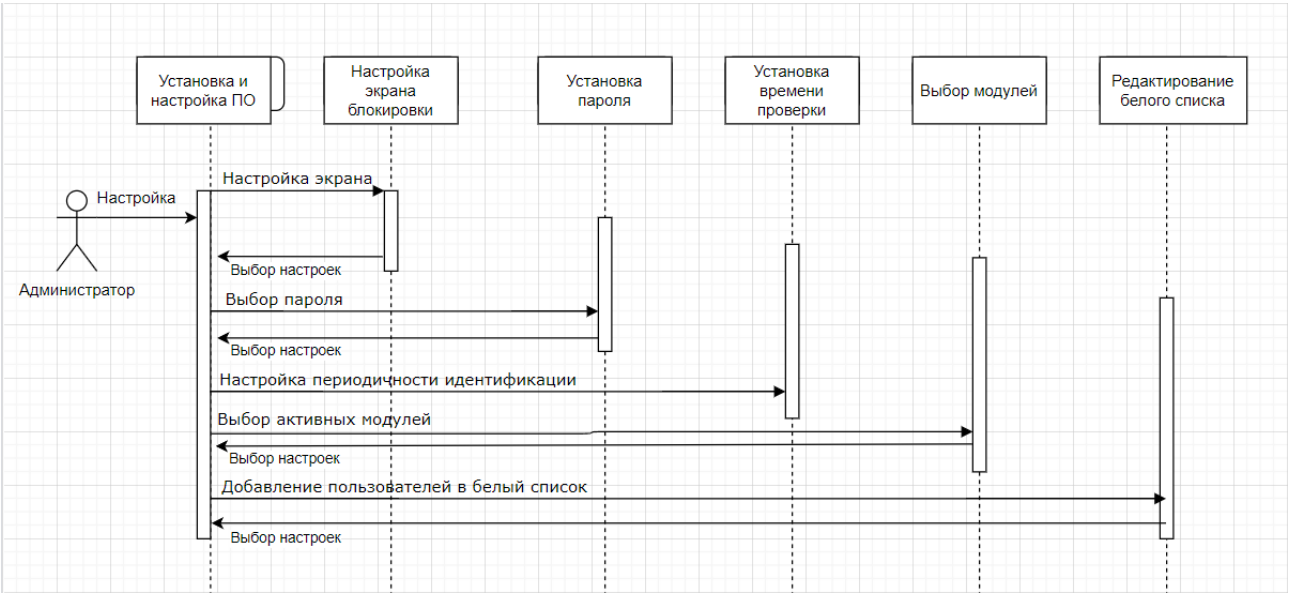

Рисунок 4 – Диаграмма последовательностей

3.1.3 Диаграмма состояний

На диаграмме (рисунок 5) представлено возможное состояние события «Настройка времени идентификации» – то есть, это то самое время, через которое будет производиться проверка идентификатора пользователя и то, присутствует ли пользователь перед камерой.

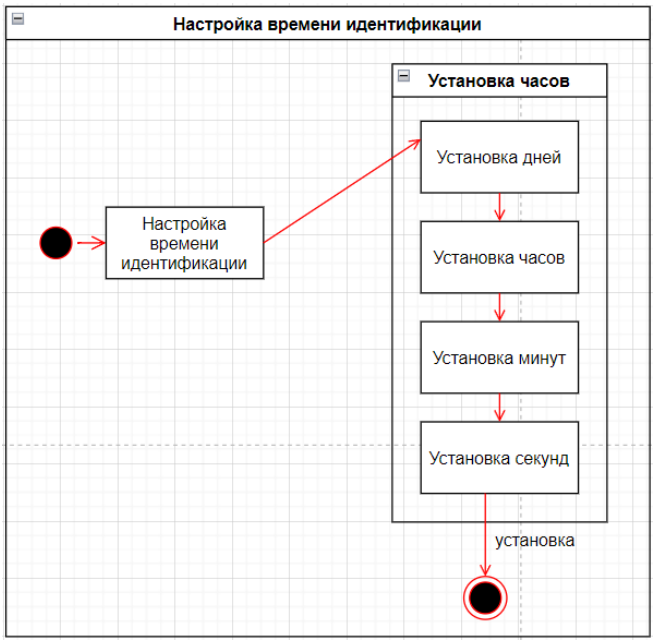

Рисунок 5 – Диаграмма состояний «Настройка времени идентификации»

3.1.4 Диаграмма деятельности

Диаграмма деятельности представляет переходы потока управления от одной деятельности к другой.

На рисунке 6, представлена диаграмма деятельности, которая описывает работу вкладке «Выбор модулей». То есть, пока не будет установлен «флажок» на одной из функций программы, то она не будет активной.

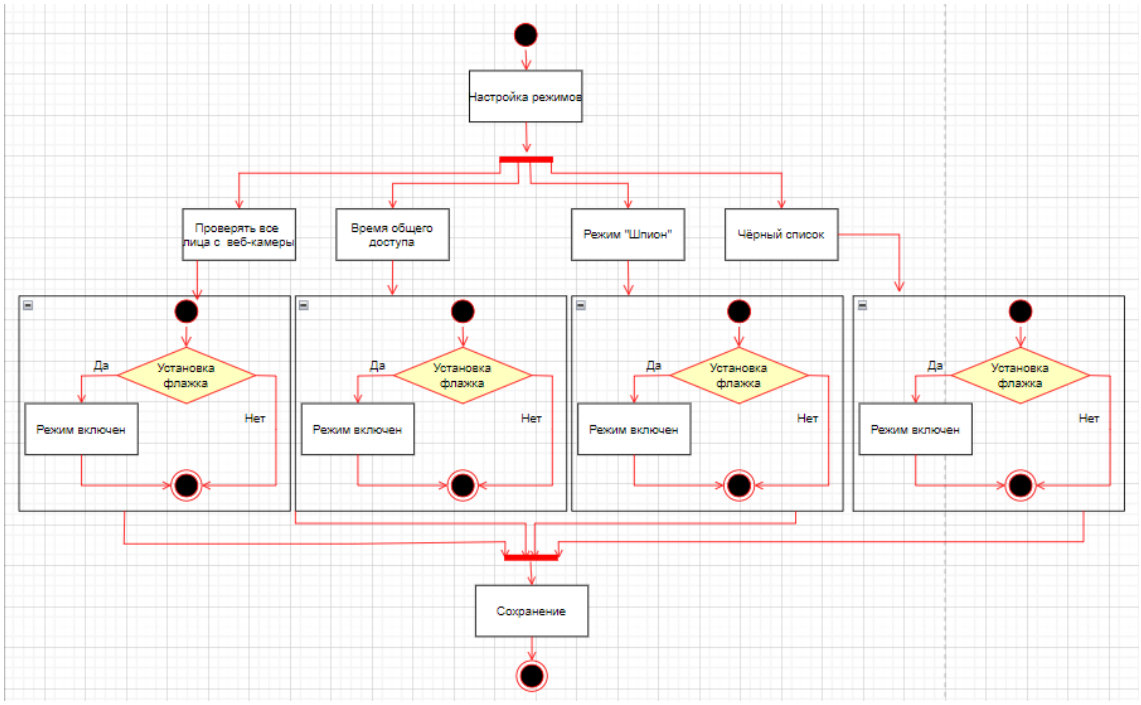

Рисунок 6 – Диаграмма деятельности «Режимы»

3.1.5 Диаграмма компонентов

Диаграмму компонентов (рисунок 7) необходимо разработать для построения концептуальной и физической схем составных частей разрабатываемой программы, так как с её помощью можно выполнить описание высокоуровневое представление о компонентах системы. Кроме того, будет спроектировано отношение множества компонентов и зависимостей между ними.

На компьютере расположена локальная база данных, в которой хранятся фотографии зарегистрированных пользователей (нужна для обеспечения функции распознавания при отсутствии интернет-подключения). FaceID, Screenshotter и удалённый сервер базы данных сообщаются между собой с помощью шины API.

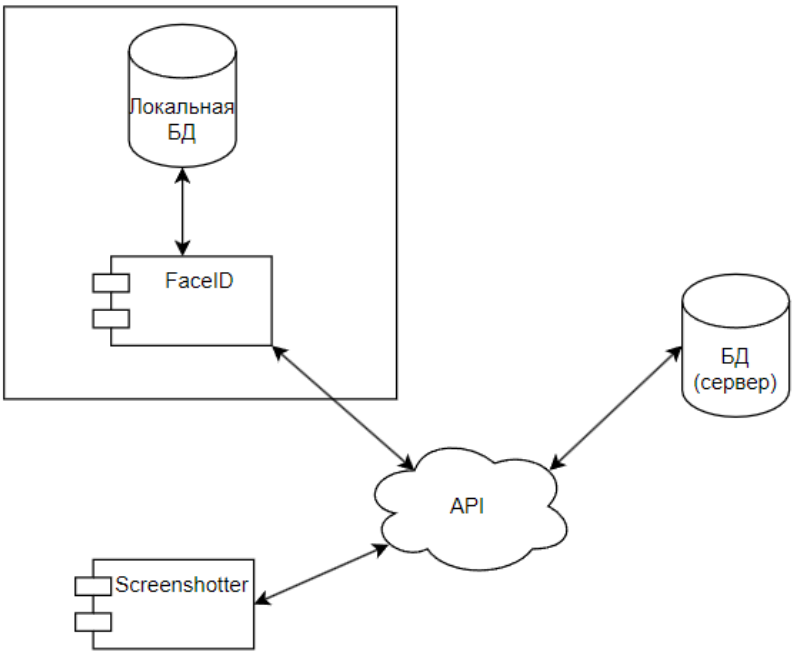

Рисунок 7 – Диаграмма компонентов

3.2 Практическая реализация информационной системы

3.2.1 Экранная форма настроек

Интерфейс администратора включает в себя функцию выбора экрана блокировки, настройки безопасности, время до следующей идентификацией, режимы работы программы, а также список разрешенных пользователей.

Пользователь компьютера, помимо администратора, не имеет доступа к функциям программы и в большинстве режимов даже не догадывается о том, что за его действиями может вестись наблюдение.

На рисунках 8-14 представлен интерфейс всех форм, доступных администратору, а именно: «Экран блокировки», «Безопасность», «Время», «Режимы» и «Белый список».

Настройки данного приложения позволяют выбрать экранную картинку для блокировки экрана (рисунок 8).

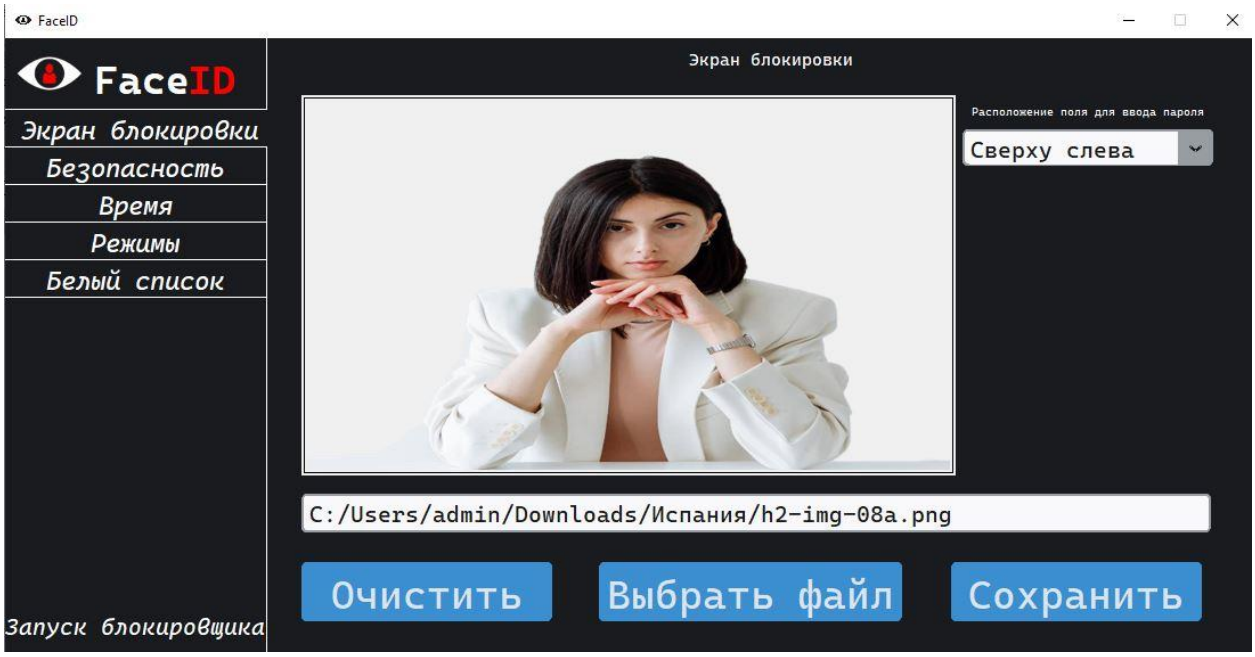

Рисунок 8 – Экран блокировки

Кнопка «Очистить» позволяет очистить поле выбора пути к файлу.

Кнопка «Выбрать файл» позволяет обратиться к проводнику компьютера и выбрать изображение из памяти компьютера.

Кнопка «Сохранить», позволяет сохранить записанный ранее или введённый вручную путь к изображению, что будет заменять экран блокировки.

Также, предусмотрена установка места, в котором будет находиться форма для ввода пароля. Данная функция позволяет отойти от стандартного расположения окна «по центру экрана» – необычное расположение усложняет доступ нарушителю (рисунок 9).

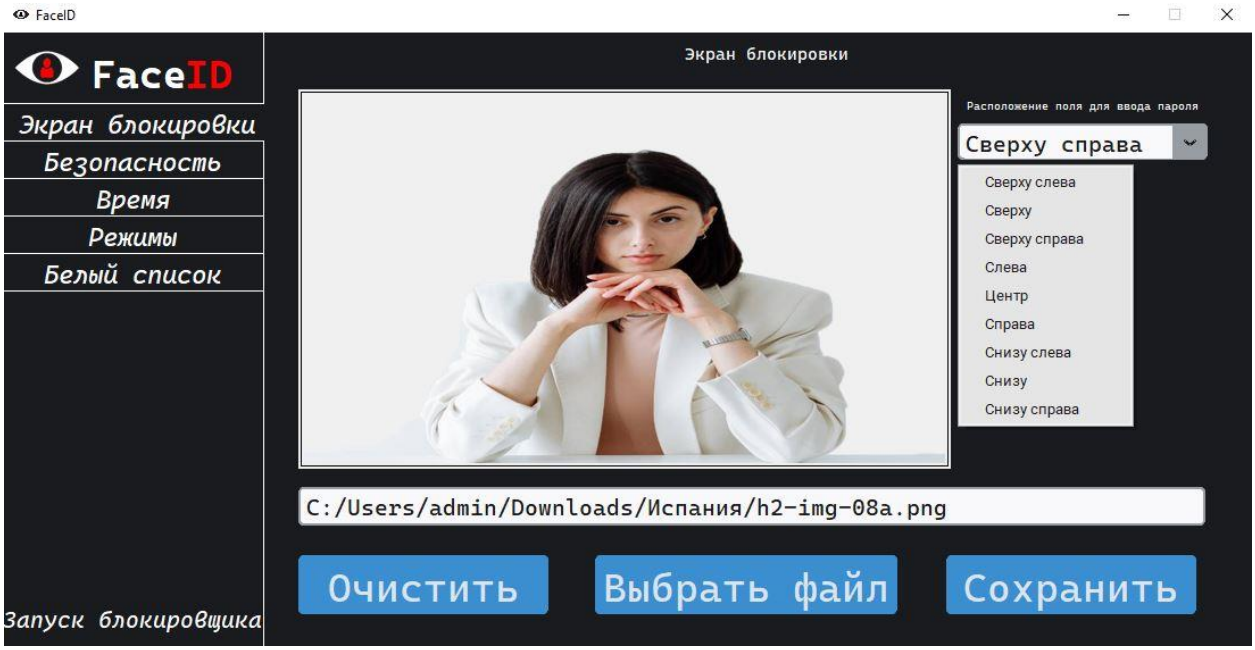

Рисунок 9 – Настройка положения пароля

На рисунке 10 представлена непосредственно форма для ввода пароля администратора. Также, можно посмотреть всех администраторов (администратор и те, кто находится в белом списке).

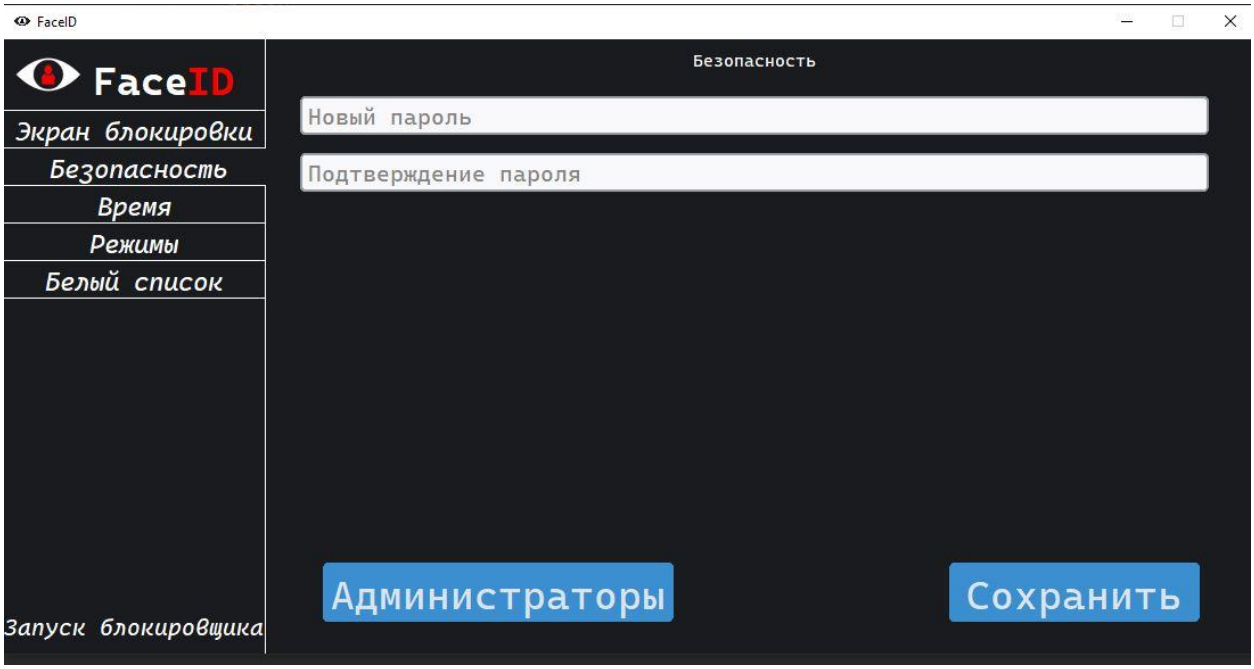

Рисунок 10 – Настройка пароля

На рисунке 11 представлено время для проверки идентификаторов пользователей.

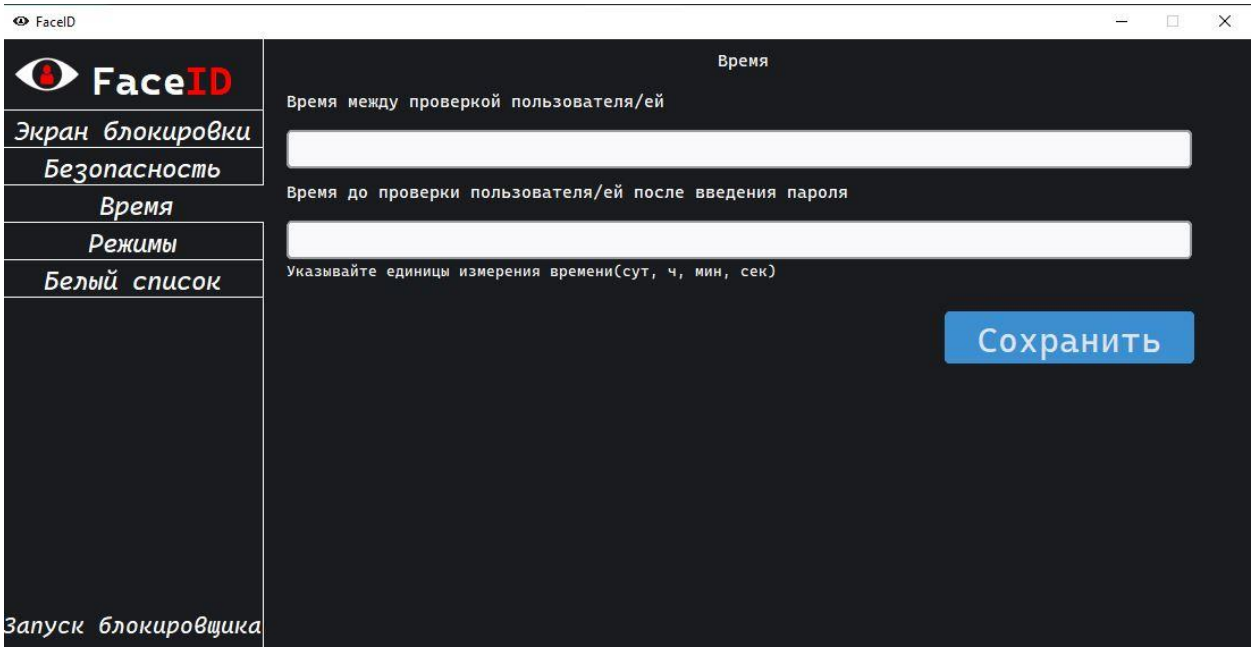

Рисунок 11 – Настройка времени проверки идентификатора

На рисунке 12 описаны возможные режимы работы программы. После выбора нужных режимов необходимо нажать кнопку «Сохранить»

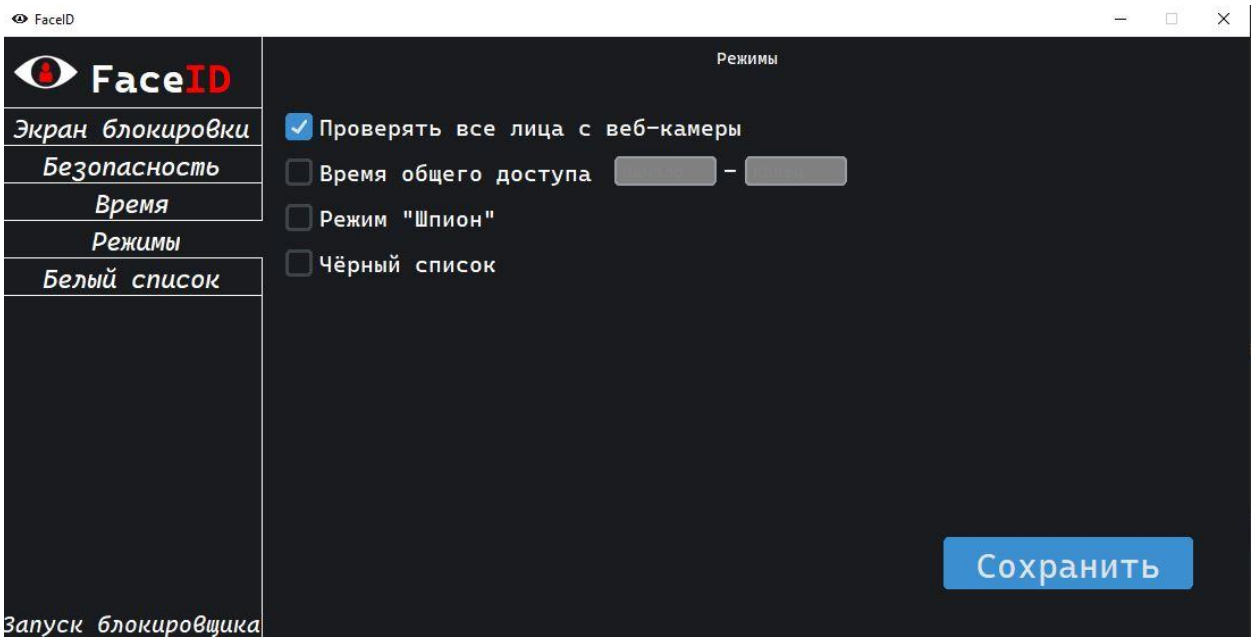

Рисунок 12 – Настройка режимов работы

На рисунке 13 представлен белый список (безусловно разрешенные пользователи для входа в программу). Белый список может быть пустым.

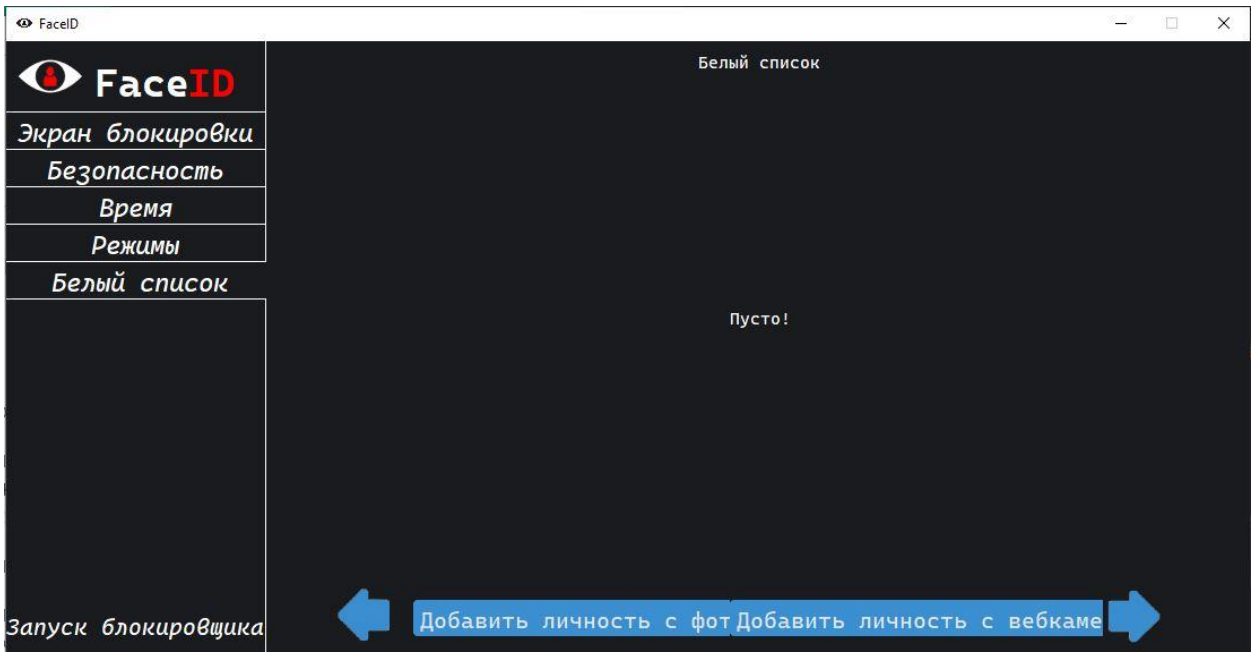

Рисунок 13 – Окно белого списка

Когда к разрешенному пользователю (администратор или пользователь из белого списка) в пределы видимости камеры попадает пользователь из черного списка, появляется надпись, уведомляющая об этом рисунок 14. Это единственная функция программы, доступная зарегистрированным пользователям.

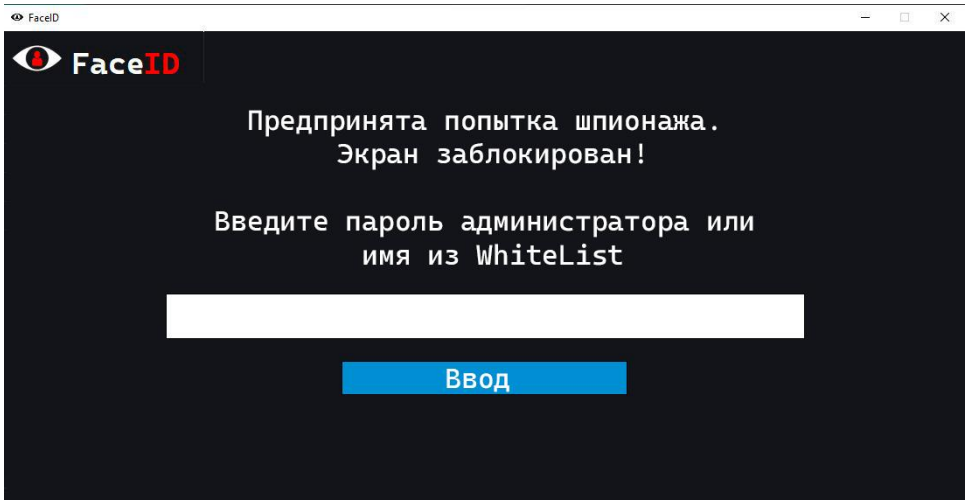

Рисунок 14 – Предупреждение о вторжении

Администратор может просмотреть какие действия совершались пользователями в течении дня, открыв файл logs.txt, который находится в одной папке с программой.

3.2.2 Telegram-бот

Для удалённого доступа к персональному компьютеру (а точнее, использования приложения через удалённый доступ) был создан Telegram-бот.

Telegram-бот создан на Python с помощью библиотеки Aiogram. Его основной функцией является слежка за неавторизованными пользователями. В моменте, когда назначается администратор, и система начинает работать, бот начинает свою работу. Это является аналогом удалённого супервизора, который может делать запись экрана и скриншоты сессии. Прервать сессию пользователя он удалённо не может, в связи с особенностью архитектуры приложения.

Практическое применение бота, заключается в том, чтобы фиксировать прецеденты безопасности для авторизованных паролем пользователей. Простой пример: администратор создал приватный текстовый файл, авторизованный паролем пользователь в системе, теоретически, не имеет к папке доступа, но практически, может открыть его и даже посмотреть содержимое. Это в реальном времени может быть зафиксировано супервизором системы, для дальнейшего разбирательства. Другими словами, помогает получить факт несанкционированного доступа к контенту.

При активации бота, а также для совершения любой операции, будет запрашиваться идентификатор – фотографию администратора, а также, в некоторых случаях, пароль администратора (рисунок 15).

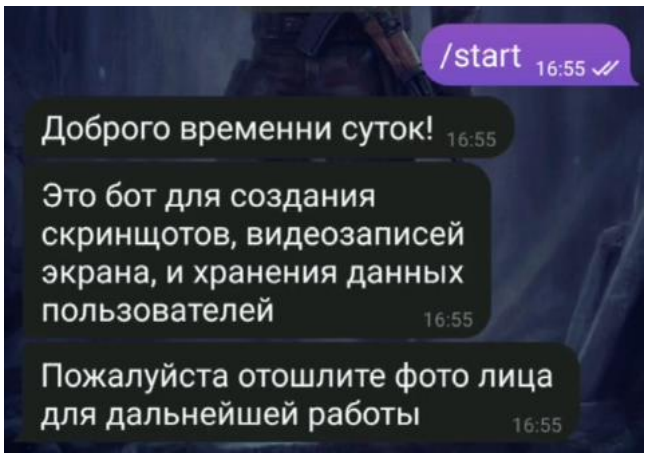

Рисунок 15 – Запрос пароля

После инициализации, вам будет доступен выбор – работать с базой данных или со снимками (видео) с экрана (рисунок 16).

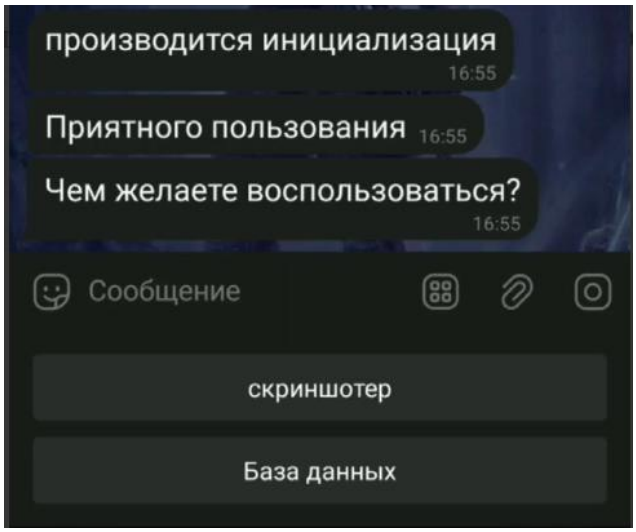

Рисунок 16 – Выбор функции

Выберем работу с базой данных – теперь нужно определиться с выбором – просмотреть таблицу пользователей, найти конкретного пользователя или редактировать какой-либо профиль (рисунок 17).

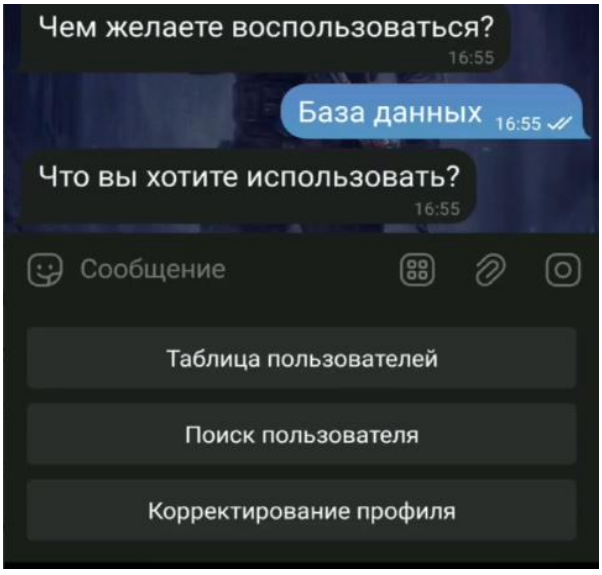

Рисунок 17 – Выбор функций из раздела «База данных»

Выберем просмотр таблицы пользователей. Для доступа необходим код администратора (рисунок 18).

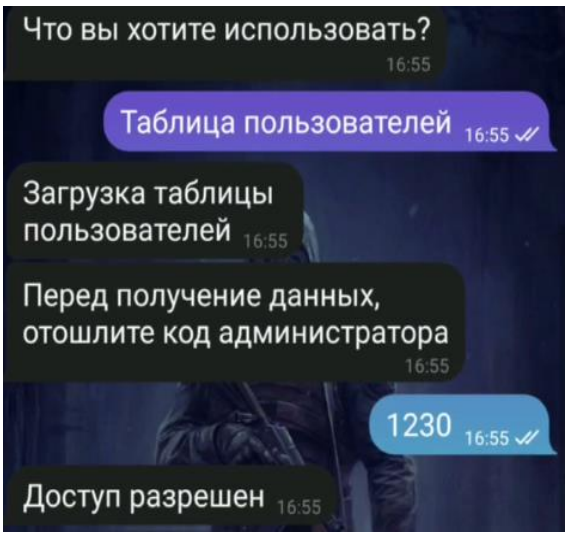

Рисунок 18 – Запрос кода администратора

После подтверждения пароля, бот вышлет базу данных (рисунок 19).

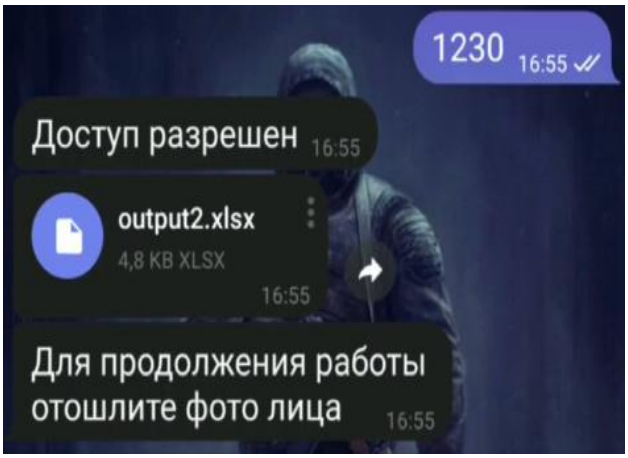

Рисунок 19 – База данных пользователей

Для использования вкладки «Поиск пользователя» алгоритм аналогичен.

Для «Корректирование профиля» алгоритм аналогичен.

В данных примерах не показан процесс отправки фотографии администратора, так как, фотография лица администратора и изображения из галереи являются личными данными, а как следствие, конфиденциальной информацией.

Если выбрать инструмент «скриншотер», то будет доступен выбор – получить видео или скриншот с экрана персонального компьютера или ноутбука, где установлено приложение FaceID (рисунок 20).

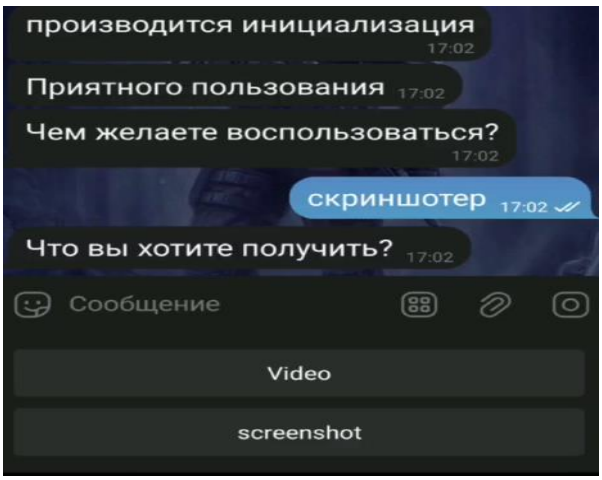

Рисунок 20 – Выбор функций из раздела «скриншотер»

Далее нужно выбрать качество изображения, если это запрос скриншота, либо длительность, если это видео (рисунок 21 и рисунок 22).

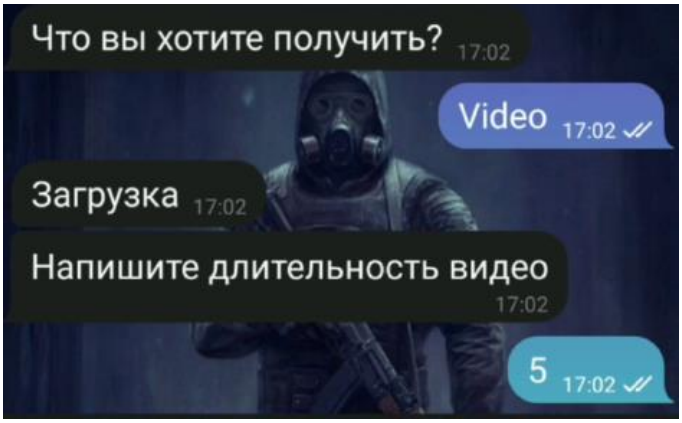

Рисунок 21 – Выбор длительности запрашиваемого видео

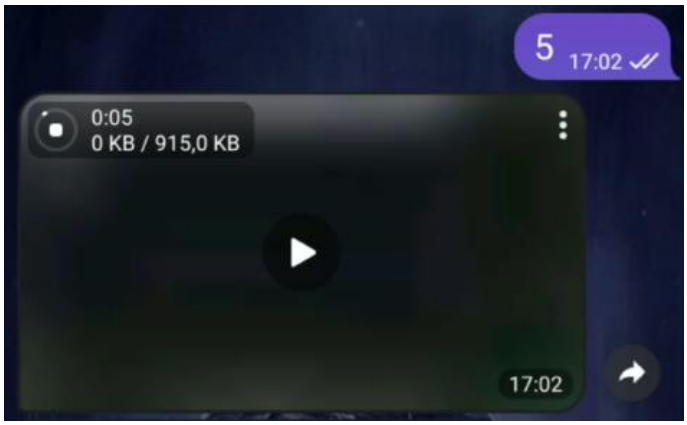

Рисунок 22 – Получение видео

Данный бот можно улучшить, добавив функцию блокировки нежелательных пользователей дистанционно или функцию просмотра изображения с вебкамеры, либо встроенной камеры компьютера.

3.3 Анализ достоверности и практической значимости результатов

3.3.1 Оценка достоверности результатов на основе сравнения с известными данными или моделями

Для оценки достоверности полученных результатов важно провести сравнение с известными данными или существующими моделями. Это позволяет проверить соответствие полученных результатов уже установленным фактам или теориям. Если результаты согласуются с известными данными или предсказаниями моделей, это повышает уверенность в достоверности результатов и подтверждает их правильность. В случае расхождений необходимо провести дополнительные исследования для выявления причин и уточнения результатов.

Проведем анализ и получим сравнительную оценку достоверности результатов.

Оценка достоверности результатов:

- Сравнение с известными программами – при проверке того же набора фото, что использовались в разработке, на другом ПО похожего функционала – получен такой-же результат. Правильность идентификатора гарантирует вход в систему, его неправильность блокирует вход.

- Проверка методологии и экспериментальных процедур – в данном способе оценки мы будем ссылаться на результаты тестирования компонентов по отдельности, а также тестировании их совместно. Также необходимо разработать и дополнительные методы тестирования. Нужно обратиться к экспериментальным данным и сравнить наши значения со значениями реальными.

- Воспроизводимость – код и методология могут быть легко воспроизведены другими исследователями или разработчиками.

В данном исследовании стоит выделить достоинства и недостатки, среди достоинств возможность простого взаимодействия и наличие открытого кода

для взаимодействия, а среди недостатков: неточность распознавания при неудачных ракурсах.

3.3.2 Оценка практической значимости результатов для решения конкретной задачи или проблемы

Для оценки практической значимости необходимо выделить сферу применения данного решения, его достоинства и недостатки и сравнить с аналогичными решениями. Практическая значимость полученных результатов определяется их способностью вносить вклад в решение конкретной задачи или проблемы. Важно оценить, каким образом результаты исследования могут быть использованы на практике и какие выгоды они могут принести.

Данная программа предназначена для идентификации пользователя. Это делает программу полезной как в сфере безопасности, так и в повседневной жизни.

3.3.3 Выводы о результатах и их влиянии на практику исследования

Анализ полученных результатов позволяет сделать выводы о их влиянии на практику исследования в образовательной сфере. Важно подчеркнуть вклад исследования в расширение знаний в определенной области, его потенциальное применение в реальной жизни и перспективы дальнейших исследований. Это помогает оценить ценность проведенного исследования и его вклад в развитие науки, технологий и практических приложений.

ЗАКЛЮЧЕНИЕ

В ходе выполнения магистерской диссертации было разработано приложение, которое идентифицирует пользователя во время его работы на персональном компьютере или ноутбуке.

В работе выполнены следующие задачи:

- проведён анализ предметной области, а именно, проанализированы методы распознавания лица, в следствии чего был подобран оптимальный способ для разработки;

- проанализированы существующие программные решения, выявлены общие функции, на основе которых может быть разработано функционально лучшее программное решение.

- рассмотрены методологии проектирования программ и выбрана та, что подходит для текущей разработки.

- разработана программа с помощью интерпретатора Python 12.1 в среде IDE Pycharm 2024.1 с использованием открытых библиотек (dlib, openCV, face_recognition и др.), которая позволяет идентифицировать пользователя во время его работы на персональном компьютере, а также блокировать доступ незарегистрированным пользователям.

- разработан Telegram-бот для удалённой работы администратора с приложением.

Возможно улучшение функционала приложения, например, добавлением функции менеджера паролей или расширением чат-бота функцией получения записи с веб-камеры или встроенной камеры устройства.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Барсук, А. А. Разработка приложения для визуальной идентификации пользователя / А. А. Барсук, Т. А. Галаган. // АмГУ, Молодёжь XXI века: шаг в будущее. – 2023. – Т. 4. С. 147 – 148.

2 Барсук, А. А. Разработка программного обеспечения «Фиксация активности пользователей ПК» / А. А. Барсук // Планирование, проведение и толкование итогов научных исследований: сборник статей Международной научно-практической конференции (20 января 2024 г., г. Киров). – Уфа: Аэтерна, 2024. – 238 с.

3 Белов, А. С. Модернизация системы информационной безопасности = Modernization of the Information Security System: The Approach to Determining the Frequency / А. С. Белов, М. М. Добрышин, Д. Е. Шугуров // Защита информации. Инсайд. – 2022. – № 4. – С. 76-80.

4 Беляев, В. В. Безопасность информационных систем: учебник для вузов / В. В. Беляев. – М. : Юнити-Дана, 2021. – 360 с.

5 Беляева, И. В. Архитектура информационных систем: учебное пособие / И. В. Беляева. – Ульяновск: УлГТУ, 2019. – 192 с.

6 Берг, Д.Б. Модели жизненного цикла : учебное пособие / Д. Б. Берг, Е. А. Ульянова, П. В. Добряк. – Екатеринбург: Изд-во Урал. ун-та, 2014. – 74 с.

7 Брэдшоу, Ш. MongoDB: полное руководство. Мощная и масштабируемая система управления базами данных / Ш. Брэдшоу. – М. : ДМК Пресс, 2020. – 540 с.

8 Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон. – М. : ДМК Пресс, 2008. – 496 с.

9 Вапник, В. Н. Теория распознавания образов / В. Н. Вапник, А. Я. Червоненкис. – М. : Наука, 1974. – 415 с.

10 Васильев, А. Н. Программирование на Python в примерах и задачах / А. Н. Васильев. – М. : Эксмо, 2021. – 619 с.

11 Вигерс, К. Разработка требований к программному обеспечению / К. Вигерс. – М. : Русская Редакция, 2004. – 576 с.

12 Внуков, А. А. Защита информации : учебное пособие для вузов / А. А. Внуков. – М. : Издательство Юрайт, 2024. — 161 с.

13 Галимянов, А. Ф. Архитектура информационных систем / А. Ф. Галимянов, Ф. А. Галимянов. – Казань: Казан. ун-т, 2019. – 117 с.

14 Горелик, А. Л. Методы распознавания / А. Л. Горелик, В. А. Скрипкин. – М. : ВШ, 1989. – 230 с.

15 ГОСТ Р ИСО/МЭК 12207-2010. «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств». – М. : Стандартинформ, 2011. – 44 с.

16 Гринченков, Д. В. Математическая логика и теория алгоритмов для программистов: учебное пособие / Д. В. Гринченков, С. И. Потоцкий. – М. : КноРус, 2017. – 206 с.

17 Громов, Ю. Ю. Информационная безопасность и защита информации: учебное пособие / Ю. Ю. Громов, В. О. Драчев, О. Г. Иванова. – Старый Оскол: ТНТ, 2017. – 384 с.

18 Гуц, А. К. Математическая логика и теория алгоритмов / А. К. Гуц. – М. : Ленанд, 2016. – 128 с.

19 Гэддис, Т. Начинаем программировать на Python. / Т. Гэддис. – СПб. : БХВ-Петербург, 2019. – 768 с.

20 Демиденко, А. Telegram Bot. Руководство по созданию бота в мессенджере Телеграм / А. Демиденко. – М. : Литрес, 2023. – 28 с.

21 Земцов, А. А. Алгоритмы распознавания лиц / А. А. Земцов. – М. : Академическое издательство ЛЭПА Ламберта, 2011. – 128 с.

22 Зенков, А. В. Информационная безопасность и защита информации : учебное пособие для вузов / А. В. Зенков. – М. : Издательство Юрайт, 2024. — 107 с.

23 Златопольский, Д. М. Основы программирования на языке Python / Д. М. Златопольский. – М. : ДМК Пресс, 2017. – 284 с.

24 Зюзьков, В. М. Математическая логика и теория алгоритмов / В. М. Зюзьков. – М. : ГЛТ, 2018. – 176 с.

25 Игошин, В. И. Теория алгоритмов: учебное пособие / В. И. Игошин. – М. : ИНФРА-М, 2016. – 318 с.

26 Интеграция корпоративных приложений с помощью ESM-системы [Электронный ресурс]. – Режим доступа: https://simpleone.ru/blog/integracziya-korporativnyh-prilozhenij-s-pomoshhyu-esm-sistemy/. – 12.06.2024.

27 Казарин, О. В. Основы информационной безопасности: надежность и безопасность программного обеспечения : учебное пособие для среднего профессионального образования / О. В. Казарин, И. Б. Шубинский. – М. : Издательство Юрайт, 2024. – 342 с.

28 Как работает распознавание лиц и можно ли обмануть эту систему – РБК. [Электронный ресурс]. Режим доступа: https://trends.rbc.ru/trends/industry/6050ac809a794712e5ef39b7?from=copy. – 12.06.2024.

29 Коул, Р. Блистательный Agile. Гибкое управление проектами с помощью Agile, Scrum и Kanban / Р. Коул, Э. Скотчер. – СПб. : Питер, 2019. – 304 с.

30 Коцюба, И. Ю. Основы проектирования информационных систем: учебное пособие / И. Ю. Коцюба, А. В. Чунаев, А. Н. Шиков. – СПб. : Университет ИТМО, 2015. – 206 с.

31 Крупский, В. Н. Математическая логика и теория алгоритмов: учебное пособие для студентов учреждений высшего проф. образования / В. Н. Крупский, В. Е. Плиско. – М. : ИЦ Академия, 2016. – 416 с.

32 Кулаков, К. И. Архитектура и фреймворки веб-приложений: учебное электронное пособие / К. А. Кулаков, В. М. Димитров. – Петрозаводск: Издательство ПетрГУ, 2020. – 61 с.

33 Кухарев, Г. А. Биометрические системы. Методы и средства идентификации личности человека / Г. А. Кухарев. – М. : Политехника, 2001. – 240 с.

34 Кумагина, Е. А. Модели жизненного цикла и технологии проектирования программного обеспечения: учебно-методическое пособие / Е. А. Кумагина, Е. А. Неймарк. – Нижний Новгород: Изд-во ННГУ, 2016. – 41 с.

35 Лебедев, А. А. Криптография и защита информации: учебное пособие / А. А. Лебедев, О. В. Морозов. – СПб. : Питер, 2019. – 320 с.

36 Малюк, А. А. Информационная безопасность: концептуальные и методологические основы защиты информации / А. А. Малюк. – М. : ГЛТ, 2016. – 280 с.

37 Фаулер, М. NoSQL: новая методология разработки нереляционных баз данных / М. Фаулер. - М. : Диалектика / Вильямс, 2016. - 921 с.

38 Мартишин, С.А. Базы данных. Практическое применение СУБД SQL- и NoSQL-типа для применения проектирования информационных систем / С.А. Мартишин. – М. : 368, 2023. – 368 с.

39 Меджуи, М. Непрерывное развитие API. Правильные решения в изменчивом технологическом ландшафте / М. Меджуи. – СПб. : Питер, 2022. – 368 с.

40 Монгоклуб. Что это за СУБД? Плюсы, минусы, подводные камни [Электронный ресурс]. Режим доступа: https://skillbox.ru/media/code/mongodb-chto-eto-za-subd-plyusy-minusy-podvodnye-kamni. – 12.06.2024

41 Мюллер, Д. П. Python для чайников / Д. П. Мюллер. – СПб. : Диалектика, 2019. – 416 с.

42 Петрова, И. Р. Методология объектно-ориентированного моделирования. Язык UML / И. Р. Петрова, Р. Х. Фахртдинов, А. А. Сулейманова, И. О. Разживин, А. Г. Фазулзянов. – Казань: Казан. ун-т, 2018. – 79 с.

43 Садаладж, Прамодкумар Дж. NoSQL. Новая методология разработки нереляционных баз данных / Прамодкумар Дж. Садаладж, Мартин Фаулер. - М. : Вильямс, 2015. - 192 с.

44 Редмонд, Э. Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL / Э. Редмонд, Д. Р. Уилсон. – М. : ДМК Пресс, 2013. – 384 с.

45 Рейтц, К. Автостопом по Python / К. Рейтц, Т. Шлюссер. – СПб. : Питер, 2017. – 336 с.

46 Федоров, Д. Ю. Программирование на языке высокого уровня Python : учеб. пособие для прикладного бакалавриата / Д. Ю. Федоров. – М. : Издательство Юрайт, 2019. – 161 с.

47 Харрисон, М. Как устроен Python / М. Харрисон. – СПб. : Прогресс книга, 2019. – 272 с.

48 Чистов, Д. В. Проектирование информационных систем: учебник и практикум для академического бакалавриата / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк– М. : Издательство Юрайт, 2018. – 258 с.

49 Что такое микросервисная архитектура? [Электронный ресурс]. – Режим доступа: https://selectel.ru/blog/what-is-microservice-architecture/. – 12.06.2024.

50 Солем, Ян Эрик. Программирование компьютерного зрения на Python / Ян Эрик Солем – М. : ДМК Пресс, 2016. – 312 с.

ПРИЛОЖЕНИЕ А

Техническое задание

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Полное наименование системы

Приложение для визуальной идентификации пользователя «Faseblock»

1.2 Наименование предприятий разработчика и заказчика системы

Разработчик: студент группы 2105-ом факультета математики и информатики Амурского государственного университета Барсук Алёна Алексеевна

Заказчик: Барсук Алёна Алексеевна

1.3 Перечень документов

Перечень документов, на основе которых проектируется система:

- ГОСТ 34.602-89 – техническое задание на проектирование автоматизированной системы управления;

- инструкция по охране труда при работе на персональном компьютере;

- первичные документы.

1.4 Плановые сроки начала и окончания работы

Плановые сроки начала и окончания работ по созданию системы: начало разработки – 01.10.2023 г., окончание – 31.05.2024 г.

2 НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

2.1 Назначение системы

Разрабатываемое приложение предназначено для фиксирование активности пользователей при использовании персонального компьютера.

2.2 Цели создания системы

Цели создания приложения:

- идентификация пользователей в процессе работы;

- фиксирование активности пользователей;

- создание отчётов об активности пользователей.

3 ТРЕБОВАНИЯ К СИСТЕМЕ

3.1 Требования к функциям, выполняемым системой

Перечень необходимых функций:

- Распознавание лица пользователя во время его работы за компьютером.

- База данных, которая хранит фото и идентификаторы на локальном и (или) на удалённом сервере.

- Приложение работает без поддержки интернета (возможна потеря некоторого функционала при отключении от всемирной сети).

- Блокировка доступа незарегистрированным пользователям во время отсутствия зарегистрированного пользователя (пользователей) у экрана.

- Реализация однопользовательского и многопользовательского режимов работы приложения.

- Фиксация действий пользователей – аналог аудита, позволяющего узнать, какие приложения использовались.

- Возможность удалённого доступа к базе данных.

- Возможность удалённого получения снимка или записи с экрана персонального компьютера или ноутбука.

- Управление базой данных – её редактирование и удаление записей.

Компоненты приложения являются взаимосвязанными. Их взаимодействие происходит в соответствии с потоками объектов и данных между ними.

В качестве среды разработки использовать программу «PyCharm».

3.2 Требования к видам обеспечения

3.2.1 Требование к математическому обеспечению

Разрабатываемая система не накладывает жестких требований к специальному математическому обеспечению.

3.2.2 Требования к информационному обеспечению

Приложение должно иметь доступ к файловой системе. Для этого необходимо выдать соответствующее разрешение.

3.2.3 Требования к лингвистическому обеспечению

Система основывается на языке программирования: Python

3.2.4 Требования к программному обеспечению

Для функционирования приложения на рабочих станциях может быть установлено любое ПО, поддерживающее язык Python

3.2.5 Требования к техническому обеспечению

Минимальные требования для работы на персональных компьютерах, имеющих следующие минимальные характеристики:

- тактовая частота процессора – 2.1 ГГц;
- ОЗУ – 2 ГБ или более;
- объем жесткого диска должен быть не менее 500 Гбайт;

К дополнительным требованиям относятся:

- устройство ввода информации: клавиатура, мышь;
- монитор;
- устройство для работы с USB Flash носителями;
- устройство захвата и записи видео;
- наличие интернет-соединения.

3.2.6 Требования к организационному обеспечению

Результат разработки ориентирован на пользователей, у которых есть персональный компьютер и им пользуется более одного человека.

Во избежание возникновения ошибок системы необходимо реализовать ограничения на вводимые параметры таким образом, чтобы не возникало неполноты данных, приводящей к возникновению конфликтных ситуаций.

3.2.7 Требования к метрологическому обеспечению

Требования к метрологическому обеспечению не предъявляются.

3.2.8 Требование к методическому обеспечению

Требования к методическому обеспечению не предъявляются.

3.3 Требования к интерфейсу

Интерфейс должен быть интуитивно понятен и требовать от пользователя минимум действий, а вся входная информация должна контролироваться во избежание ввода ошибочных и некорректных данных. По возможности, использовать в интерфейсе программы диалоговые окна, которые применяются в приложениях разработанных на Python.

3.4 Требования к эргономике и технической эстетике:

Интерфейс программы должен быть интуитивно понятен и требовать от пользователя минимум действий, а вся входная информация должна контролироваться во избежание ввода ошибочных и некорректных данных.

Система должна иметь человеко-машинный интерфейс, удовлетворяющий следующим требованиям:

- взаимодействие системы и пользователя должно осуществляться на русском языке, за исключением системных сообщений, не подлежащих русификации;

- допустима видимость предоставляемой информации на экране;

- допустимая цветопередача.

3.5 Требования к надежности и безопасности

Надежность закладывается в архитектуре системы. Она определяет, как часто происходят сбои компонентов. Требования к надежности технических средств системы должны обеспечивать возможность ее круглосуточной эксплуатации. Система должна обладать способностью восстанавливаемости

3.6 Требования к защите информации от несанкционированного доступа

Компоненты подсистемы защиты от НСД должны обеспечивать проверку полномочий пользователя при работе с системой.

3.7 Требования по сохранности информации при авариях

Система должна восстанавливать свое функционирование при корректном перезапуске аппаратных средств.

Должна быть предусмотрена возможность организации автоматического и (или) ручного резервного копирования данных системы средствами системного и базового ПО (ОС, СУБД), входящего в состав программно-технического комплекса.

4 СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМЫ

4.1 Перечень стадий и этапов работ по созданию системы

Этапы, которые необходимо выполнить по созданию приложения для персонального компьютера:

1 этап – исследование предметной области, анализ документации, выделение объекта автоматизации. По окончании данного этапа будут разработаны контекстные диаграммы, диаграммы потоков данных и другие схемы.

2 этап – составление технического задания: выяснение требований заказчика к разрабатываемой системе, определение технических и программных средств, необходимых для реализации проекта, уточнение функций системы.

3 этап – разработка рабочей документации на систему.

4 этап – разработка программного продукта с использованием языка Python.

5 этап – программная реализация приложения.

6 этап – согласование программной реализации приложения с требованиями заказчика с учетом всех замечаний и пожеланий.

7 этап – внедрение и сопровождение приложения: установка и настрой-ка программного средства, обучение пользователей работе с приложением, выявление и устранение неполадок.

4.2 Сроки выполнения

На разработку приложения отводится срок с 29.01.2024 по 10.06.2024.

4.3 Состав организации исполнителя работ

Все работы выполняются студентом Амурского государственного университета Барсук Алёной Алексеевной.

4.4 Вид и порядок экспертизы технической документации

Вид и порядок экспертизы технической документации определяет Заказчик в одностороннем порядке.

5 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ СИСТЕМЫ

5.1 Виды, состав, объем и методы испытания

Приемка готовой информационной системы осуществляется по следующему плану:

1 этап – анализ готового проекта;

3 этап – выполнение корректировки и дополнения системы по результатам предыдущих этапов;

4 этап – составление списка достоинств и недостатков спроектированной системы

5.2 Общие требования приемки работ по стадиям

Сдача-приёмка работ производится поэтапно, в соответствии с рабочей программой и календарным планом. Приемка осуществляется комиссией, в состав которой входят представители Заказчика. Приемка информационной системы осуществляется в присутствии представителей Исполнителя. По результатам приемки подписывается акт приемочной комиссии.

Все создаваемые в рамках настоящей работы программные изделия передаются Заказчику, как в виде готовых модулей, так и в виде исходных кодов, представляемых в электронной форме на стандартном машинном носителе.

6 ТРЕБОВАНИЯ К СОСТАВУ И СОДЕРЖАНИЮ РАБОТ ПО ПОДГОТОВКЕ ОБЪЕКТА АВТОМАТИЗАЦИИ К ВВОДУ СИСТЕМЫ В ДЕЙСТВИЕ

6.1 Преобразование входной информации к машиночитаемому виду

Вся исходная информация, используемая в проектируемой подсистеме, должна быть приведена к виду, пригодному для обработки в ЭВМ. На этапе ввода в эксплуатацию первичное информационное наполнение информационной подсистемы должно соответствовать ее функциональному назначению.

6.2 Сроки и порядок комплектования и обучения персонала

Заказчику необходимо до начала работ по созданию подсистемы сформировать штат специалистов в обязанности, которых будет входить контроль над ходом создании подсистемы, а также утвердить штат персонала, который будет являться непосредственными пользователями и администраторами разрабатываемой информационной системы. Сроки, программы обучения и состав групп должны быть определены на этапе подготовки и разработки и могут в дальнейшем уточняться.