

Содержание

Введение	3
1 Обзор предметной области	4
1.1 История развития СКУД	4
1.1.1 Первое поколение	4
1.1.2 Второе поколение	5
1.1.3 Третье поколение	6
1.2 Основные составляющие СКУД	7
1.2.1 Контроллер	7
1.2.2 Считыватель	8
1.2.3 Программное обеспечение	9
1.3 Виды СКУД	10
2 Обзор и анализ существующих решений	12
3 Обзор платформы для разработки и проектирование клиентской части	16
3.1 Требования и задачи	16
3.2 Аппаратная часть СКУД	16
3.3 Программное обеспечение СКУД	19
3.3.1 Чат-бот	20
3.3.2 Модуль распознавания лиц	20
3.3.3 Модуль управления реле	21
3.4 Выбор инструментов разработки	21
3.4.1 Работа с камерой	21
3.4.2 Работа с распознаванием лиц	22
3.4.3 Построение чат-бота	22
3.4.4 Выводы по разделу 3	23
4 Разработка системы контроля и управления доступом	25
4.1 Подготовка и настройка платформы СКУД	25
4.1.1 Выбор и установка операционной системы	25
4.1.2 Подготовка операционной системы для разработки	26
4.2 Структура проекта	27
4.3 Модуль распознавания лиц	28
4.3.1 Детектирование лиц со статичного изображения	28
4.3.2 Распознавание лиц со статического изображения	29
4.3.3 Тренировка распознавания системы	31
4.3.4 Распознавание лиц с видеопотока	32
4.4 Модуль Telegram-бот	34

Изм.	Лист	№ докум.	Подп.	Дата	Тема	Код специальности		
						Лит.	Лист	Листов
Разраб.	Имя автора							
Провер.	Имя проверяющего							
Реценз.								
Н. Контр.								
Утверд.								
						Группа	1	40

4.4.1	Получение и отправка данных Telegram-ботом	34
4.4.2	Построение пользовательских меню	35
4.4.3	Построение диалога с пользователем	35
5	Экономическая часть	37
6	Охрана труда и экология	38
	Заключение	39
	Список литературы	40

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						2

Введение

Система контроля и управления доступом (далее СКУД) - это совокупность программных и аппаратных средств, предоставляющая возможность управления пропускным режимом, с целью ограничить доступ к определённым территориям или помещениям лицам, не имеющим к ним разрешения. Подобные системы широко используются среди не только крупных организаций и предприятий, но и малого бизнеса, индивидуальных предпринимателей. Причинами такого успеха являются:

- централизованное управление пропускным режимом на объекты;
- сокращение времени на проверку документов;
- упрощение ведения статистики.

Для обеспечения контроля доступа в больших предприятий существует большое количество решений на рынке, однако для малых помещений с соответственно пониженной ценой наблюдается недостаток предложений. Именно поэтому, до сегодняшнего дня, малые организации используют простые замки или, к примеру, домофоны. Такие устройства значительно снижают удобство и быстроту доступа к определенным объектам.

Решением данной проблемы является разработка собственной СКУД, что определяет цель дипломного проекта. Для достижения данной цели должны быть решены следующие задачи:

- изучение теории по системам СКУД;
- анализ существующих решений на рынке, выявление их основных возможностей и недостатков;
- изучение теории по методам идентификации пользователей;
- выбор программных и аппаратных инструментов для разработки системы СКУД.

Структура работы включает в себя ...

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						3

1. Обзор предметной области

СКУД - система контроля и управления доступом. В более широком смысле это совокупность программных и аппаратных средств технической защиты, цель которых - ограничение и отслеживание событий входа/выхода на определенной территории.

1.1. История развития СКУД

История СКУД начинается в связи с потребностью введения контроля над доступом людей на ограниченные территории, заменив старые, неактуальные способы контроля на автоматизированные и удобные в управлении системы. Структурная схема первых систем (на сегодняшний день также используются):

- считыватель (программно-аппаратное устройство, которое принимает коды от внешних устройств);
- валидатор (логический блок, предназначенный для проверки кода на его соответствие);
- реле (программный или аппаратный модуль, предназначенный для управления устройствами блокировки прохода).

1.1.1. Первое поколение

СКУД первого поколения выполняли лишь базовые функции: считыватель ключей получал определенный код, передавал его в валидатор, далее валидатор проверял код на соответствие и принимал решение о открытии/закрытии блокирующего устройства.

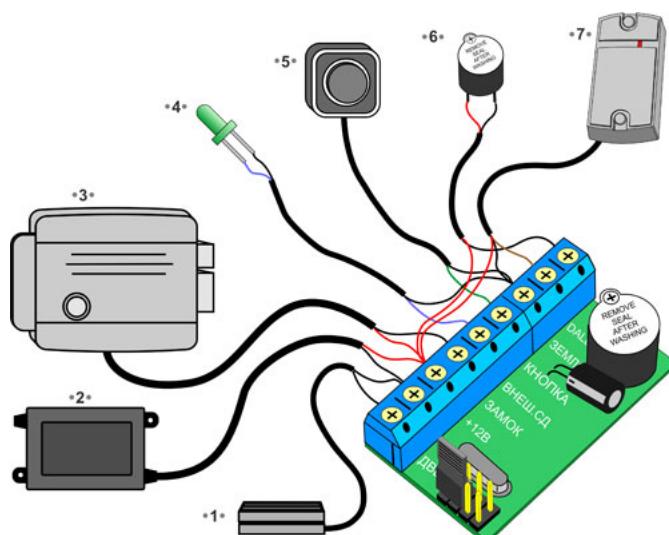


Рисунок 1 – пример простого СКУД – Z-5R

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						4

В качестве устройств содержащих в себе ключ доступа широко использовалась TouchMemory – устройства, имеющие однопроводный протокол обмена информацией и флеш-память для её хранения.



Рисунок 2 – электронный ключ Button.com, реализующий систему TouchMemory

Подобные устройства хоть и не отличались большой функциональностью, внесли в развитие СКУД несколько нововведений:

- отказ от линий связи, что привело к использованию палиативных механизмов программирования;
- внедрение протокола MicroLan, на основе которого позже будет создано множество охранных и пожарных систем.

Несмотря на все достоинства, СКУД первого поколения не отвечали еще некоторым основным требованиям – ведению журнала событий и простоте программирования контроллеров.

1.1.2. Второе поколение

Опыт использования первых СКУД показал, что они востребованы, однако пользователи нуждаются в увеличении функциональности и безопасности.

В связи с увеличением мощности микроконтроллеров, а также появлением доступа к технологиям Ethernet. Из этого исходят следующие нововведения:

- появились журналы событий. Они не работали в реальном времени, но значительно повышали безопасность охраняемых объектов;
- контроллеры стали способны сами получать и передавать данные для обработки, без необходимости в управляющих командах;
- на платах СКУД предусматривался канал связи Ethernet. Отныне локальные сети стали ключевыми линиями связи.

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						5

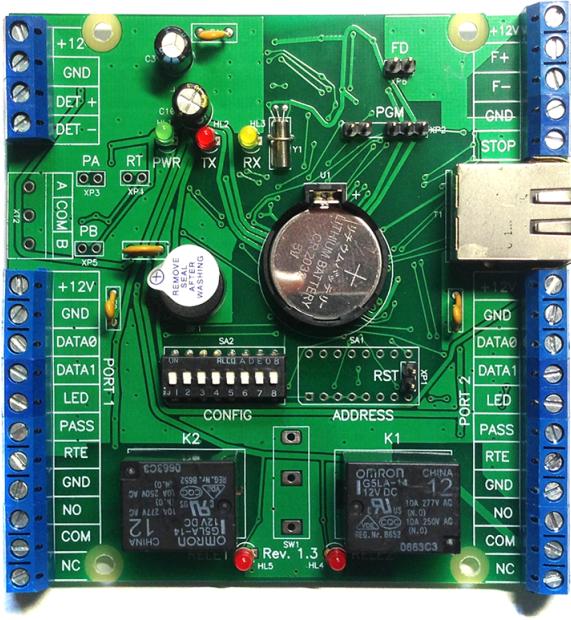


Рисунок 3 – Сетевая СКУД Sigur – поддерживает работу по сети за счет встроенного Ethernet порта

В процессе использования СКУД нового поколения, были выявлены следующие недостатки:

- журналы событий не работали в реальном времени. Такая функция позволила бы незамедлительно реагировать на определенные события;
- не было поддержки более сложных алгоритмов валидации ключей, соответственно безопасность нельзя было повысить;
- не поддерживались разные виды идентификаторов, такие как штрих-коды, отпечатки пальцев и т.п.

1.1.3. Третье поколение

Применение локальных сетей ускорило развитие СКУД. Ранее совмещенные элементы систем контроля и управления доступом стали разделяться, образовывая распределенные системы. Это означало, что стало возможным располагать на разные объекты считыватели, подключённые к одному контроллеру СКУД.

Также, чаще роль контроллера СКУД стал выполнять персональный компьютер, с установленным управляющим программным обеспечением. Такая структура позволила:

- наделить системы СКУД гибкостью, возможностью использования различного ПО, выбираемого по требуемым характеристикам;

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						6

- использовать любые поддерживающие методы идентификации – от простых TouchMemory до биометрии.

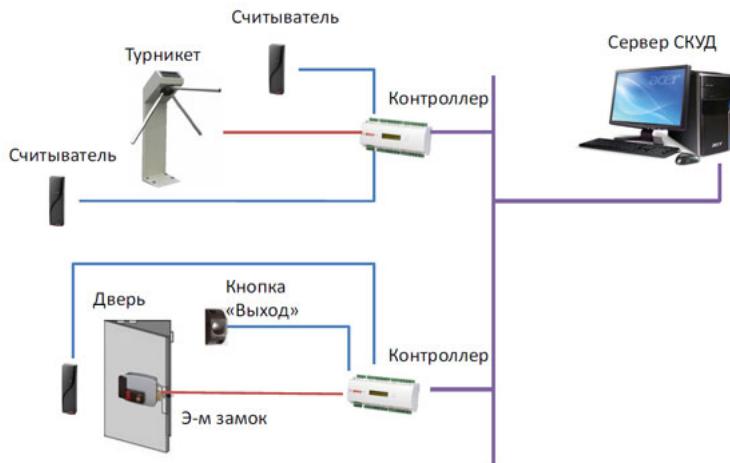


Рисунок 4 – структурная схема СКУД третьего поколения

Третье поколение СКУД вследствие больших нововведений получила и ряд нерешённых проблем:

- все элементы системы теперь зависели от одного-двух серверов, на которых выполнялось управляющее программное обеспечение;
- за счёт большого числа новых методов идентификации появилось множество проблем с безопасностью. Такая ситуация подтолкнула производителей к использованию и разработке новых протоколов обмена информацией между устройствами.

1.2. Основные составляющие СКУД

1.2.1. Контроллер

В современных системах контроля и управления доступом контроллер – одна из самых функциональных модулей. Он выполняет роль центрального модуля управления – в его памяти хранятся коды идентификаторов, а значит контроллер принимает решение о допуске определённого человека на охраняемый объект.

В случае необходимости автономного контроллера, он совмещается со считывателем в одном устройстве, что позволяет сократить затраты, снизить стоимость и упростить монтаж.

При использовании СКУД на несколько точек доступа, ощутимо возрастает значимость характеристик.

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						7



Рисунок 5 – пример контролеров – автономный Z-5R (слева), Anviz SAC844 (справа)

1.2.2. Считыватель

Считыватель - устройство, которое получает код идентификатора и передаёт его в контроллер для обработки. Считыватели, в зависимости от модели, позволяют принимать в качестве идентификатора следующие идентификаторы:

- proximity-карты;
- TouchMemory;
- код доступа;
- биометрия (отпечатки пальцев, радужная оболочка глаз, лица).

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

Код специальности

Лист



Рисунок 6 – считыватели Smartec

1.2.3. Программное обеспечение

В случае использования компьютера в качестве контроллера, существует специальное программное обеспечение, роль которого – управление подключёнными модулями и хранение идентификаторов.

Программное обеспечение используется при необходимости более широкого функционала, чем может предоставить обычный контроллер, например:

- ведение отчётности, быстрый доступ к отчётам;
- подключение баз данных сотрудников, например в 1С;
- удалённое управление, изменение баз данных, конфигураций.

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						9

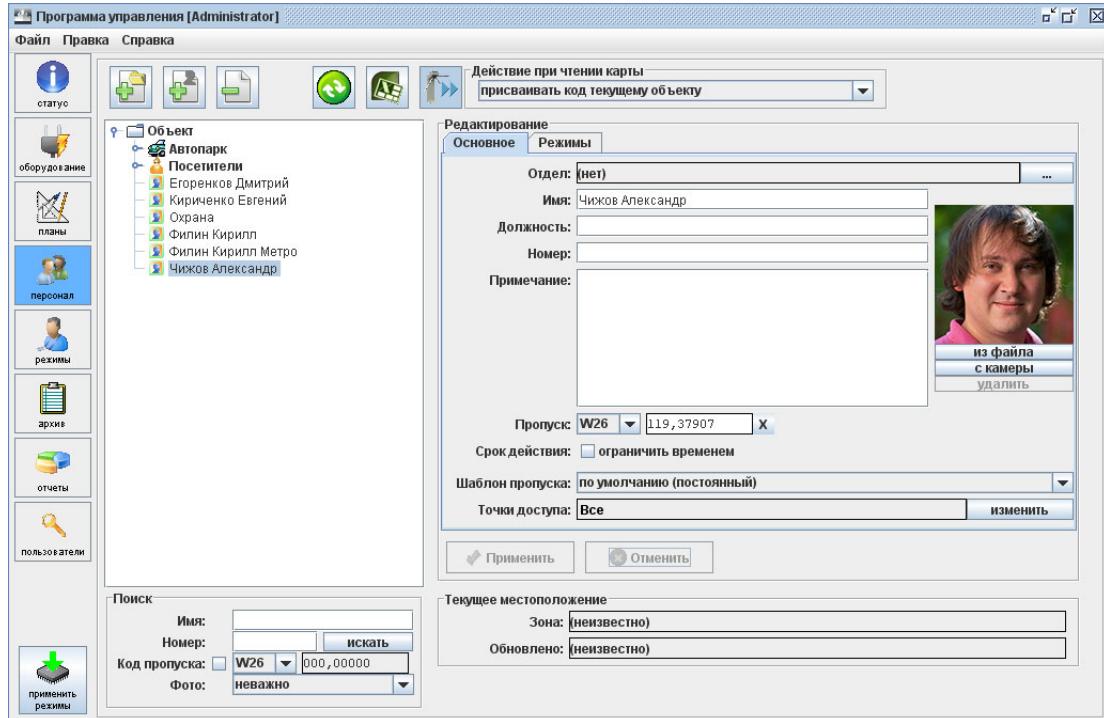


Рисунок 7 – программное обеспечение Castle

1.3. Виды СКУД

Все системы контроля и управления доступом можно разделить на две категории: сетевые и автономные системы.

В сетевой системе все контроллеры соединены с компьютером, что позволяет управлять десятками дверей, проходных пунктов, турникетов. Подобные системы удобны для больших объектов (офисов, производственных предприятий).

Сетевые системы используются для:

- использования сложных алгоритмов допуска сотрудников с разными привилегиями в разные зоны объекта;
- организации учёта рабочего времени;
- при взаимодействии с другими системами безопасности, например с пожарной сигнализацией.

В сетевой СКУД могут применяться как проводные, так и беспроводные методы передачи данных, например:

- Bluetooth;
- Wi-Fi;
- GSM.

Изм.	Лист	№ докум.	Подп.	Дата

Автономные системы менее функциональны, дешевле, проще в эксплуатации. Они не требуют прокладки сотен метров кабеля, а также сопряжения и управления с компьютера. При этом, автономные системы могут иметь некоторый функционал сетевых СКУД, например, ведение отчётов, удалённое управление, но должны обеспечивать безопасность хранения информации, т.к. все идентификаторы располагаются непосредственно в автономной системе.

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						11

2. Обзор и анализ существующих решений

В качестве примеров существующих на рынке СКУД, были выбраны следующие системы:

- комплексная система с распознаванием лиц Sigur;
- Smartec ST-FR040EM;
- комплексная система Perco.

Важно отметить, что для выполнения сравнения системы Sigur и Perco были теоретически собраны из представленных модулей.

Краткое описание всех выбранных систем:

В продуктах Sigur распознавание лиц используется для автоматической идентификации сотрудников в точках прохода. Видеопоток может быть получен с IP-камер, подключённых к системе напрямую.

Комплексная система Sigur поддерживает как только распознавание лиц, так и другие методы идентификации совместно.

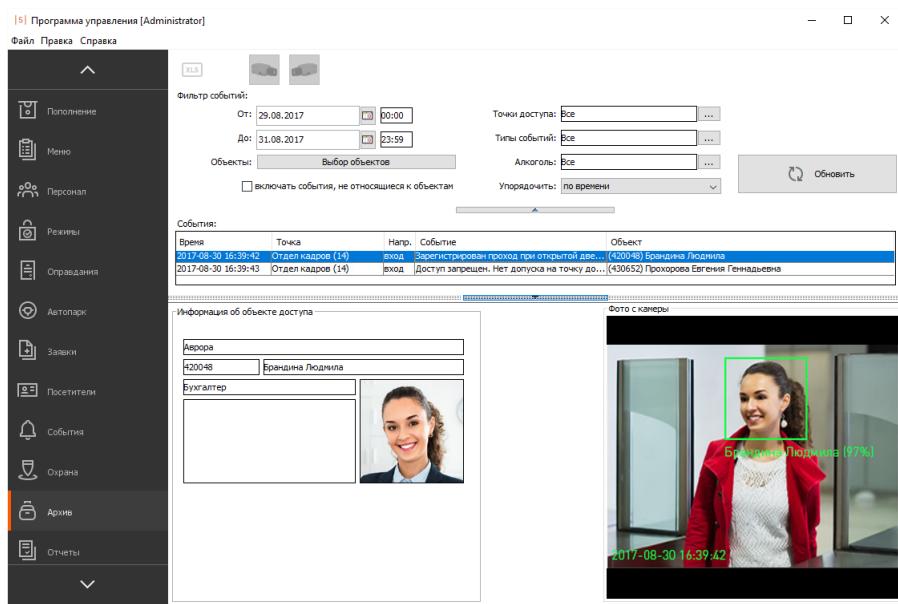


Рисунок 8 – СКУД Sigur

Корректная работа гарантируется только при следующих условиях:

- количество лиц в базе до 1000;
- качественный видеопоток;
- качественное освещение.

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						12

При этом СКУД Sigur требует для работы компьютер/сервер со следующими техническими характеристиками: процессор Intel Core i7, не менее 8 Гб ОЗУ.

Цена продукта определения посредством лицензирования – выбором покупателем количества камер и максимального количества сотрудников в базе. Например, минимальная конфигурация, состоящая из:

- одна камера – 7000 руб.
- 10 лиц сотрудников в базе – 72000 руб.
- контроллер – 16170 руб.

стоит 95170 руб.

Следующая система, биометрический считыватель ST-FR040EM марки Smartec выполняет распознавание геометрии лица, а также идентификацию пользователей по коду доступа и картам стандарта Em Marine. Наличие встроенного контроллера позволяет ему выполнять функции СКУД в автономном режиме.



Рисунок 9 – СКУД Smartec ST-FR040EM

На передней панели считывателя под небольшим углом к вертикальной плоскости расположены две камеры: обычная цветная и камера, фиксирующая изображение в ИК-диапазоне. При этом распознавание лиц выполняется с помощью обработки кадров, зафиксированных ИК-камерой.

Помимо биометрического ридера, ST-FR040EM имеет встроенный считыватель карт доступа формата Em Marine, а также оснащен экранной кодо-наборной клавиатурой. Благодаря этому, устройство позволяет реализовать не только распознавание лиц, но и осуществлять контроль доступа по картам,

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						13

коду, а также использовать эти методы идентификации в различных комбинациях.

ST-FR040EM имеет релейный выход с НЗ/НР контактами, через который считыватель может управлять электрическим дверным замком. НР-контакт применяется в тех случаях, когда устройство для распознавания лиц должно управлять замком, открывающимся при подаче напряжения, а НЗ – когда замок открывается при отключении питания. При этом, если рабочее напряжение замка составляет 12 В, то для него и устройства распознавания можно использовать единый источник электропитания достаточной мощности. Система обеспечивает корректную работу при выполнении следующих условий:

- до 500 лиц в базе распознавания;
- температурах работы от 0 до 50 С;
- питание не более 400 мА.

Розничная цена СКУД Smartec составляет 29000 руб.

Системы контроля доступа PERCo-S-20 интегрированы с биометрическими контроллерами Suprema предназначены для учета отпечатков пальцев сотрудников и посетителей. В качестве идентификаторов в системе могут использоваться карты доступа и отпечатки пальцев совместно или по отдельности.



Рисунок 10 – биометрический СКУД Perco Suprema

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						14

Для функционирования оборудования Suprema необходимо приобрести, как минимум, Базовое ПО PERCo-SN01 и один контроллер PERCo с интерфейсом связи по Ethernet.

Система поддерживает хранение до 10 отпечатков для одного сотрудника/посетителя, а максимальное количество отпечатков, хранимых в СКУД – 20000.

При использовании контроллеров и считывателей Suprema могут быть использованы как недельные, так и сменные графики доступа для сотрудников.

Интеграция настольных считывателей серии BioMini позволяет регистрировать биометрические данные сотрудников/посетителей централизованно, например, сотрудником отдела кадра или бюро пропусков.

Цена подобной системы составляет 22470 руб.

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

Код специальности

Лист

15

3. Обзор платформы для разработки и проектирование клиентской части

3.1. Требования и задачи

Для создания системы контроля и управления доступом были сформулированы наиболее важные идеи, задачи и требования на основе проведённого теоретического анализа и обзора существующих решений.

Разрабатываемая система должна:

- осуществлять контроль доступа на основе биометрического метода идентификации – распознавания лиц. Такой метод является наиболее практичным для небольшого количества сотрудников, т.к. позволяет не приобретать карты доступа, а также даёт возможность более быстрой идентификации.
- иметь реле для управления различными видами замков, турникетов, и т.п.
- иметь клиентскую часть, представленную в виде чат-бота в мессенджере. Исследование рынка показало, что методы управления СКУД неудобны, не имеют большого функционала и часто требуют непосредственного нахождения рядом с управляющим блоком. Основные модули, из которых должна состоять клиентская часть: модуль обработки видеопотока и распознавания лиц; модуль чат-бота мессенджера. В качестве мессенджера был выбран Telegram, так как имеет самое функциональное и документированное API для создания чат-ботов и сами чат-боты Telegram отличаются от всех мессенджеров наличием таких особенностей как: возможность общения с ботом посредством клавиатур, команд (например, /help или /menu), обычных сообщений и картинок; возможность построения многоуровневых меню, что решает проблему незнания пользователем команд управления; модуль управления реле. Модуль отвечает за принятие команд от модуля распознавания, подачей напряжения на контакты GPIO для управления реле.
- обеспечивать безопасность хранения информации.

3.2. Аппаратная часть СКУД

Разработка системы контроля и управления доступом осуществляется на платформе компании Raspberry Pi Foundations – Raspberry Pi 3B. Основой данного продукта является процессор с ARM-архитектурой Cortex-A53 с частотой 1,2 ГГц и модуль оперативной памяти на 1 Гб. Raspberry Pi 3B

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						16

разработана с интерфейсами Ethernet и USB, что позволяет использовать многочисленные устройства расширения.

Поддержка технологий Wi-Fi и Bluetooth обеспечивает широкие возможности для организации соединения и контактирования внешних устройств с платой.

Raspberry Pi имеет контакты GPIO, что может быть использовано для программного управления различными устройствами.

Выбранное устройство имеет малые габариты – 85,6×53,98×17 мм. Это подразумевает то, что конечное устройство будет занимать минимальное количество места.

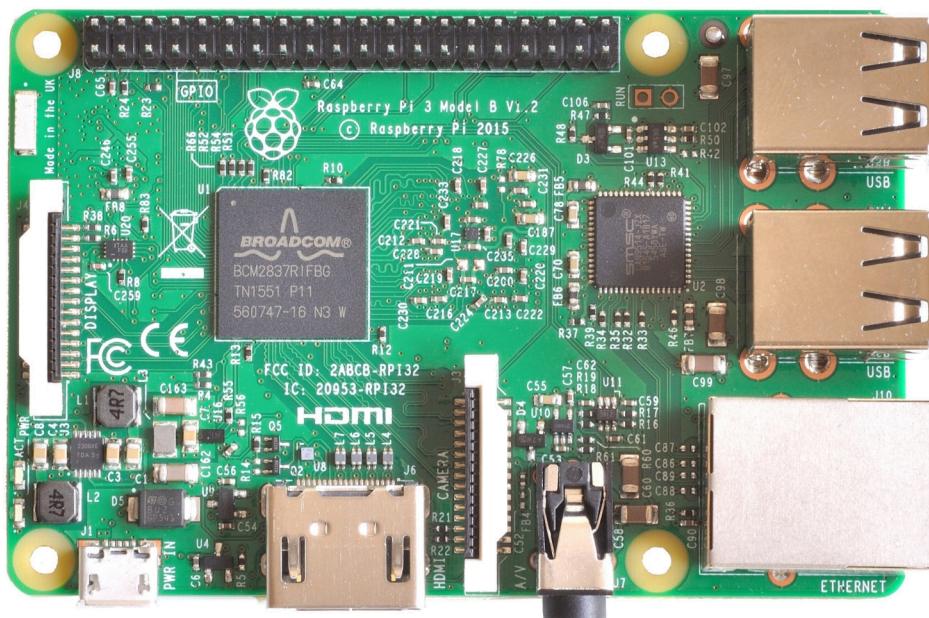


Рисунок 11 – Raspberry Pi 3B

Raspberry Pi имеет слот CSI, представляющий интерфейс между платой и модулем камеры. Такая возможность позволяет передавать данные с камеры со скоростью до 5 Гбит/с и не использовать IP-камеры, имеющие задержку из-за передачи данных по сети.

Для создания видеопотока для обработки следует использовать камеру. Благодаря описанному разъёму CSI плата может использовать модуль камеры с соответствующим интерфейсом.

В качестве модуля камеры был выбран Raspberry Pi Camera Board v2.1 – модуль от производителя основной платы. Это обеспечивает совместимость камеры с компьютером, а также даёт возможность использовать официальный SDK для работы с камерой, тем самым получая наилучшее быстродействие.

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						17

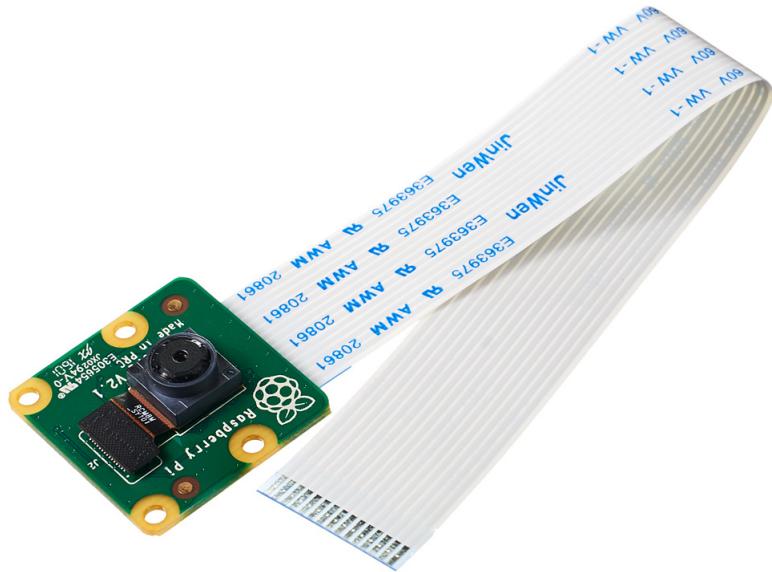


Рисунок 12 – модуль камеры Raspberry Pi Camera Board

Модуль камеры основан на сенсоре Sony IMX 219 PQ, имеет разрешение до 8 Мп (3280x2464) и поддерживает видеоформаты от 480р (90 FPS) до 1080р (30 FPS).

Для выполнения одной из главных функций устройства – управления внешними блокирующими устройствами – выбран модуль реле Tongling с рабочим напряжением 5В. Данный модуль поддерживает проходящий ток до 10А, а также имеет специальные контакты для подключения к контактам GPIO.

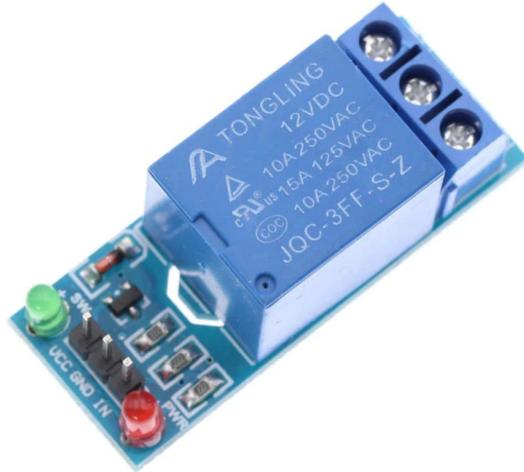


Рисунок 13 – модуль реле Tongling на 5В

Raspberry Pi не имеет стандартных разъёмов для подключения жестких дисков, но содержит слот для SD-карты. Такое решение было сделано для большей компактности. Для в качестве хранилища информации была выбрана SD-карта название.

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						18

Общая архитектура аппаратной части системы контроля и управления доступом:

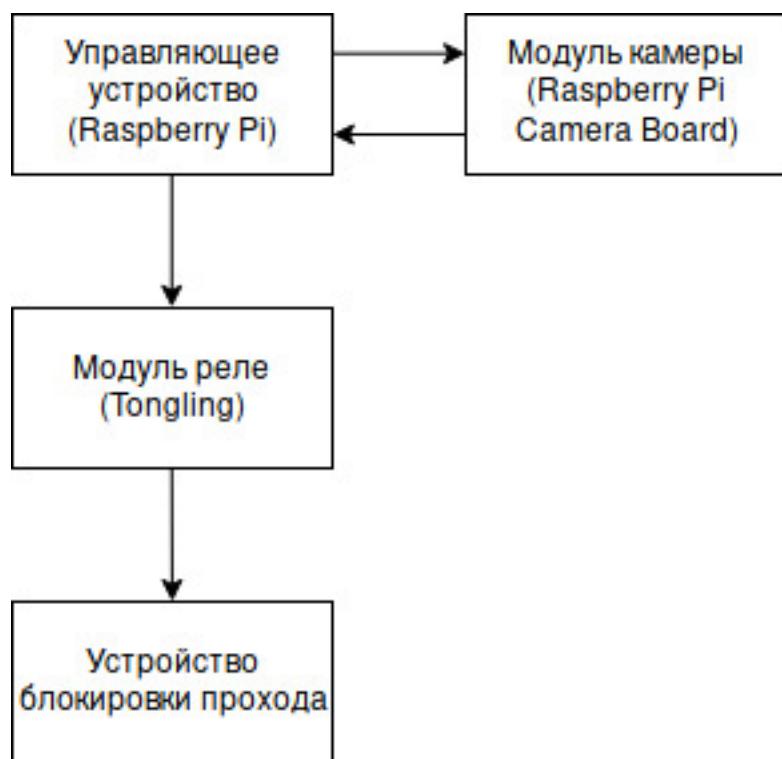


Рисунок 14 – структура аппаратной части системы

3.3. Программное обеспечение СКУД

Разработка программного обеспечения СКУД осуществляется на основе трёх модулей: чат-бота, модуля распознавания лиц и модуля управления реле.

Основная схема программного обеспечения:

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						19



Рисунок 15 – структура программной части СКУД

3.3.1. Чат-бот

Задача данного программного модуля состоит в получении пользовательских команд, обработки, конструирования и отправки ответов.

Основная функциональность, выполняемая модулем:

- построение пользовательских меню;
- построение диалогов с пользователем;
- проведение аутентификации пользователя чат-бота;
- управление базой лиц: добавление, удаление, модификация;
- уведомления о происходящих событиях пользователя (например проход распознанного/нераспознанного лица).

3.3.2. Модуль распознавания лиц

Модуль распознавания лиц выполняет одну из главных задач системы контроля и управления доступом – идентификацию пользователей. Функциональность модуля:

- обработка и подготовка изображения для детектирования лиц;
- детектирование лица на фотографии, составление выделяющих масок;

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						20

- распознавание лиц, идентификация лиц с заранее заданными фотографиями пользователей в режиме единичной фотографии или в режиме потока кадров;
- принятие решения об подаче сигнала открытия/закрытия на реле.

3.3.3. Модуль управления реле

Данный модуль представляет простое сокрытие реализации работы с контактами GPIO, то есть предоставляет интерфейс для управления состояниями реле.

Функциональность данного модуля состоит из:

- подачи сигнала отключения реле;
- подачи сигнала включения реле.

3.4. Выбор инструментов разработки

В рамках разработки системы контроля и управления доступом выбор инструментов зависит не только от возможностей тех или иных библиотек конкретных языков программирования, но и от задач поставленных в разделе 3.1.

В результате проведения обзора высокогородневых языков, в качестве которых были выбраны C++, Java, Python, был сделан выбор в пользу Python по следующим причинам:

- код программы выполняется интерпретатором, что даёт возможность запускать программу без процесса компиляции. Это ускоряет процесс разработки и отладки программного обеспечения;
- интерпретатор Python занимает гораздо меньший объём памяти по сравнению с JRE;
- большая кроссплатформенность в отличие от C++;

3.4.1. Работа с камерой

В качестве решения для работы с модулем камеры была выбрана официальная библиотека для Raspberry Pi Camera Board – picamera. Она предоставляет интерфейс для камеры на языке программирования Python.

picamera позволяет выполнять такие операции как:

- работа с изображением с камеры, сохранение, обработка (поворот, изменение цвета и т.п.);

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						21

- получение видеопотока, возможность выбора формата и количества кадров в секунду.

3.4.2. Работа с распознаванием лиц

Для распознавания лиц был выбран инструмент Face Recognition, который представляет из себя библиотеку, имеющую интерфейс на языке Python. Данная библиотека использует два основных инструмента:

- библиотеку OpenCV – широко распространённый набор алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов. Включает в себя базовые структуры, вычисления (математические функции, генераторы случайных чисел), обработку изображений, модели машинного обучения, модули для работы с калибровкой камеры.
- библиотеку dlib, для работы с нейронными сетями, машинным обучением и пр. Использует обученные каскады для поиска лиц.

Используя возможности детектирования лиц и обученных нейросетей Face Recognition добивается максимального процента распознавания лиц.

3.4.3. Построение чат-бота

Для разработки чат-бота в мессенджере Telegram была использована имплементация API Telegram на языке Python – pyTelegramBotAPI. Одна из ключевых возможностей инструмента – использование паттерна проектирования Decorator, предназначенного динамического подключения дополнительного поведения к объекту.

Основные возможности pyTelegramBotAPI:

- получение, обработка и отправка нескольких видов сообщений и медиафайлов – текстовых сообщений, аудио, видео, документов различных форматов, голосовых сообщений, стикеров и т.д.;
- управлять сообщениями, т.е. отправлять, получать, удалять, редактировать;
- генерировать меню и inline-меню (Рисунок 16)
- использовать механизм web-hooks;
- использовать любые прокси-серверы;
- собирать отчёты о работе API.

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						22

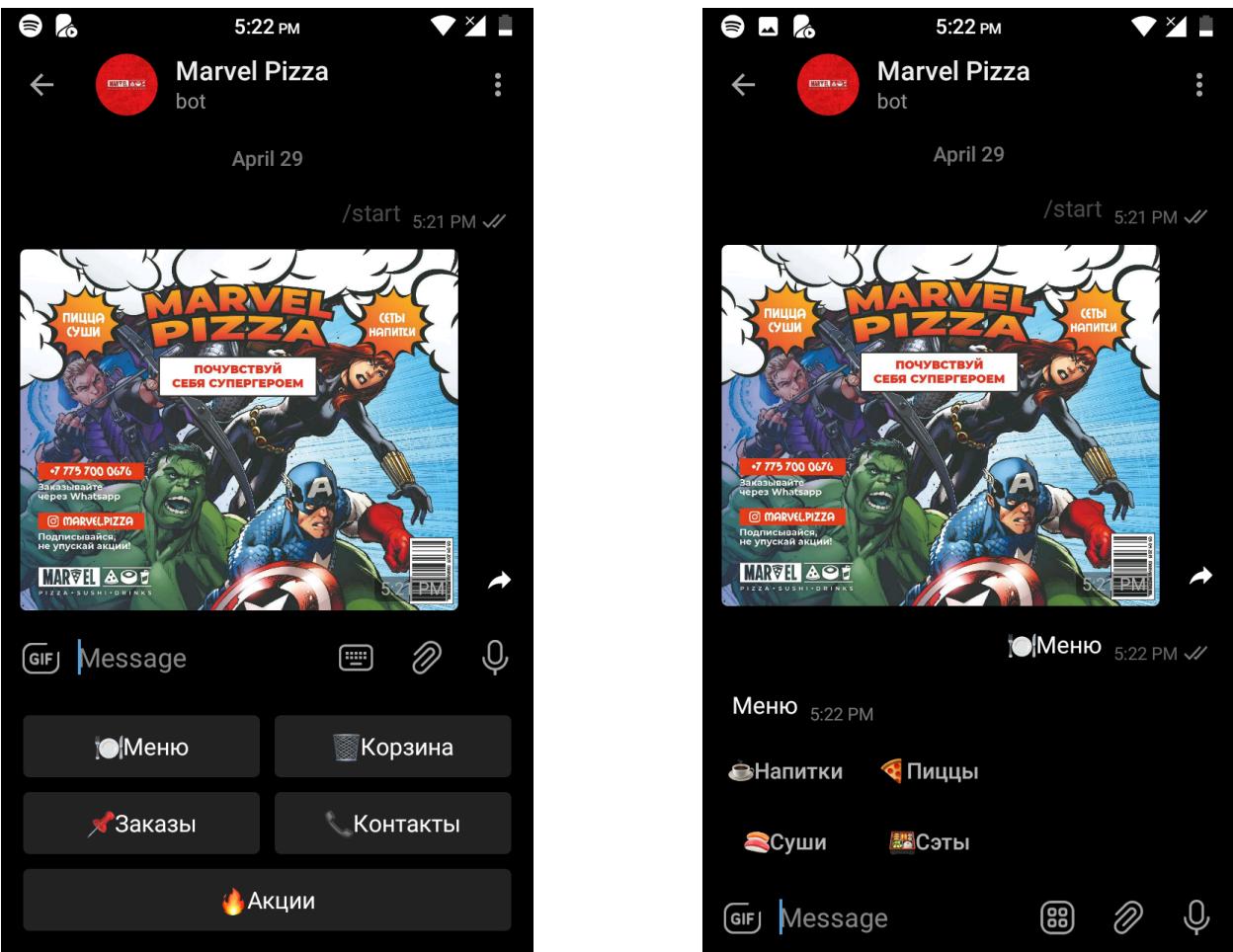


Рисунок 16 – пример меню (слева) и inline-меню (справа)

3.4.4. Выводы по разделу 3

Проведён обзор платформы для разработки СКУД. Были выявлены основные требования к аппаратным и программным модулям системы. В результате были спроектированы аппаратная и программная части системы, перечислены основные используемые устройства и библиотеки.

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						23

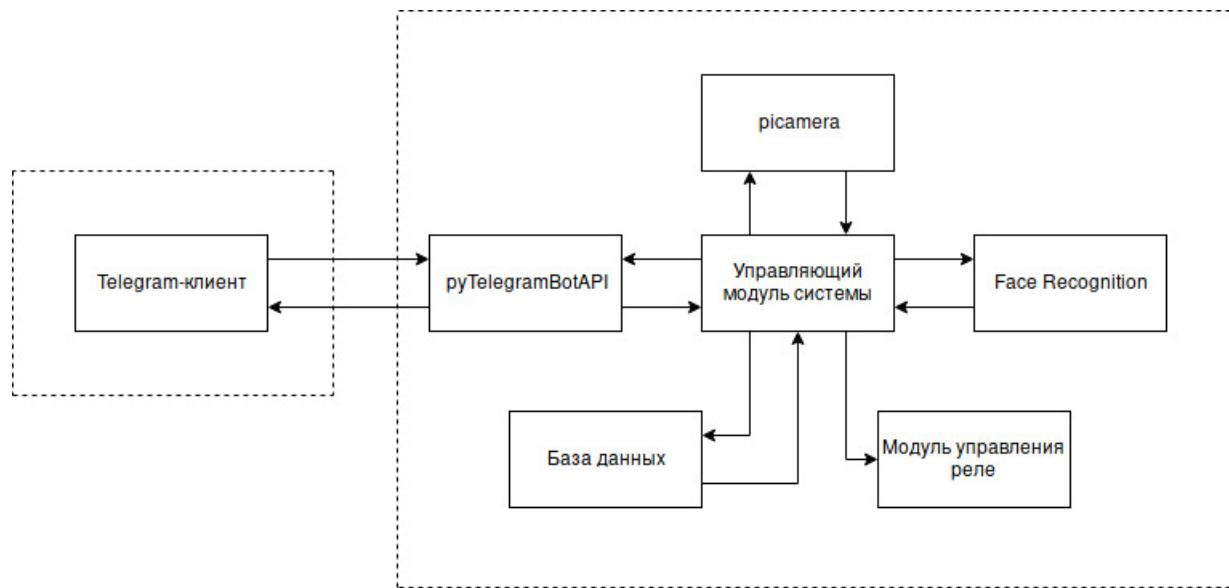


Рисунок 17 – Уточнённая структура программного обеспечения СКУД

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						24

4. Разработка системы контроля и управления доступом

4.1. Подготовка и настройка платформы СКУД

Аппаратной основой системы контроля и управления доступом является мини-компьютер Raspberry Pi 3B. Для обеспечения работы всех аппаратных и программных модулей СКУД требуется произвести выбор и установку дистрибутива операционной системы, её конфигурирование, настройку разъёмов, проводных и беспроводных подключений.

4.1.1. Выбор и установка операционной системы

Выбор операционной системы в первую очередь зависит от характеристик системы. Raspberry Pi 3B имеет процессор с ARM-архитектурой (Advanced RISC Machine) ARMv7. Такие процессоры часто требуют меньшее количество транзисторов, чем с архитектурой CISC, что позволяет уменьшить цену, потребление электроэнергии и тепловыделение. Характеристики процессоров ARM отлично подходят для легких, портативных устройств, например смартфонов, нетбуков, планшетов.

Архитектура ARM поддерживает 64-битный набор инструкций начиная с ARMv8, поэтому для установки на Raspberry Pi требуется операционная система, поддерживающая 32-битные ARM процессоры.

Под критерии подходят довольно большое количество операционных систем, так как сообщество Raspberry Pi активно их разрабатывает под самые различные задачи, например Open Source Media Center – предназначен для создания домашнего киноцентра, или Pi MusicBox – основан на Mopidy Music Streaming Server, используется для проигрывания музыки из Spotify, Google Music и т.п.

Но большинство из таких операционных систем предназначены для решения конкретной задачи и имеют предустановленные тематические приложения. Для разработки СКУД требуется операционная система имеющая инструменты для работы с контактами GPIO, поддерживающая все необходимые для разработки программные пакеты.

Также операционные системы подходящие для Raspberry Pi делятся на несколько групп:

- Unix-like системы (основанные на Linux, BSD и пр.);
- разрабатываемые компанией Microsoft (Microsoft Windows);
- разрабатываемые компанией Google Chrome OS и Android.

Для выполнения проекта была выбрана операционная система, разрабатываемая непосредственно для Raspberry Pi – Raspbian. Raspbian – это операционная система на основе ядра Linux, являющаяся потомком системы

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						25

Debian. Данный дистрибутив включает в себя множество полезных инструментов для разработки – интерпретатор Python, редактор кода, пакеты для работы с камерой и контактами GPIO.

Установка дистрибутива операционной системы тривиальна – требуется записать ISO образ дистрибутива Raspbian на карту памяти и загрузить операционную систему.

4.1.2. Подготовка операционной системы для разработки

После установки операционной системы была проведена её базовая настройка, а именно:

- настройка сетевого подключения по технологии Wi-Fi, а также протокола SSH – он позволяет управлять устройством удалённо и не запускать при этом графический интерфейс;
- настройка статического IP-адреса для подключения по SSH;
- изменение часового пояса системы.
- обновление установленных пакетов и системы с помощью пакетного менеджера Apt.

После настройки был получен доступ к эмулятору терминала Raspberry Pi по протоколу SSH.

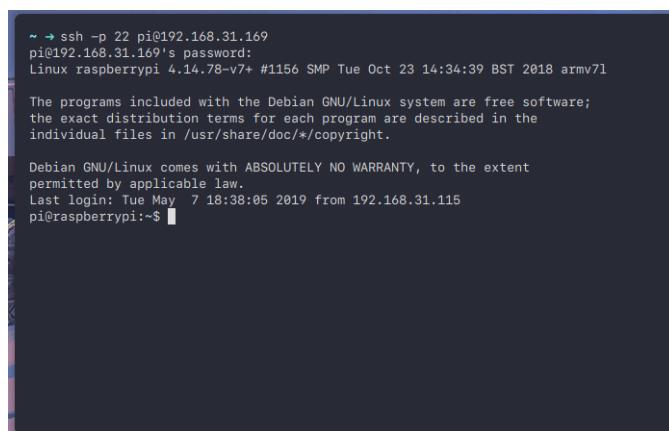


Рисунок 18 – управление Raspberry Pi по SSH

Apt - пакетный менеджер для дистрибутивов Debian. По завершению настройки операционной системы с помощью Apt были установлены необходимые для разработки программные пакеты:

- dlib;
- Face Recognition;
- OpenCV;

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						26

- pyTelegramBot;

Пакеты интерпретатора языка Python предустановлены в операционной системе Debian, поэтому их установка не требуется.

4.2. Структура проекта

Исходный код приложения представляет из себя Python—проект, разделённый на логические модули. Такая организация проекта позволяет держать исходный код в одном репозитории и упростить сборку Python приложения.

Исходный код и используемые в проекте файлы содержатся в корневой папке *smartcamera*:

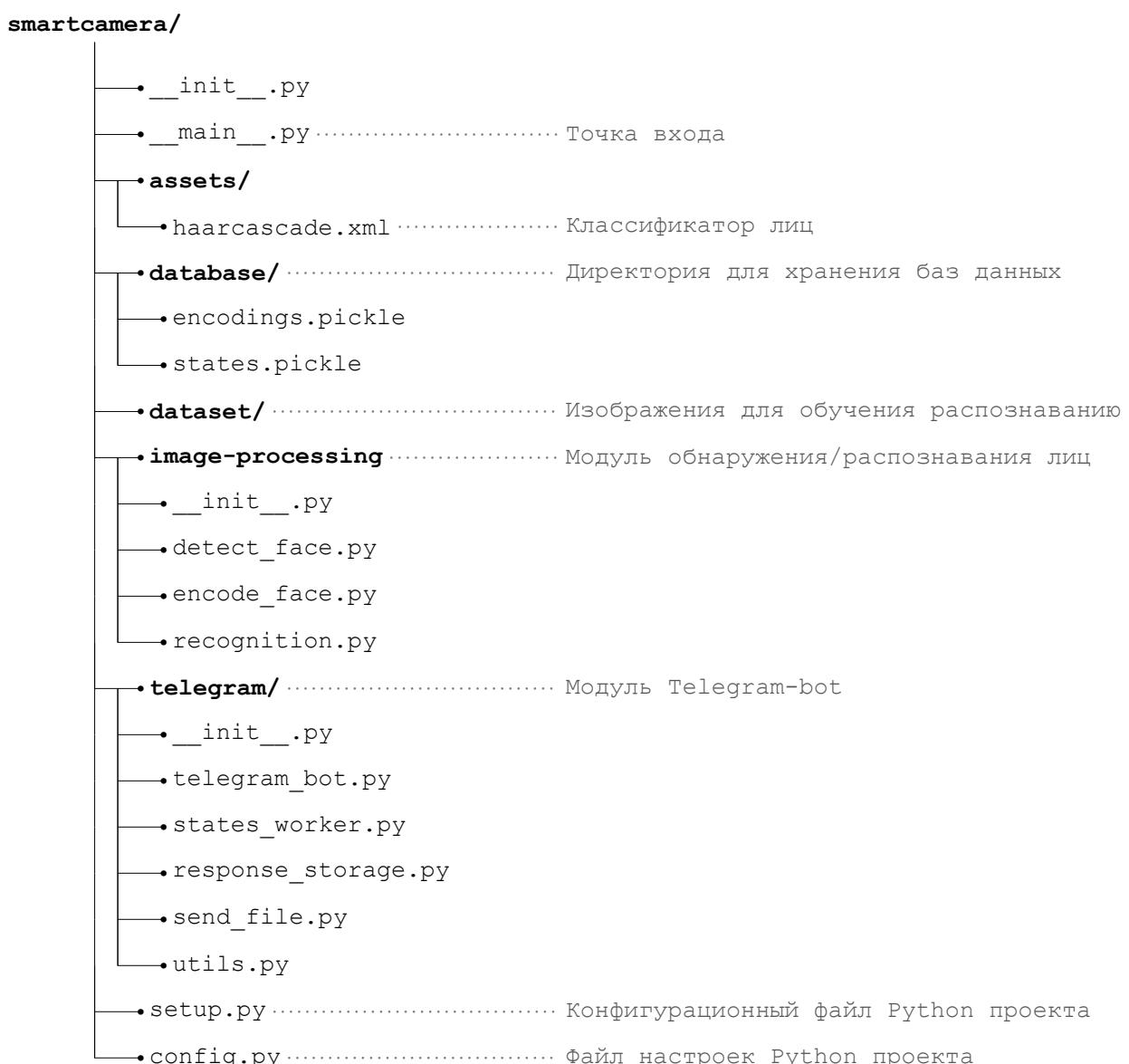


Рисунок 19 – Структура проекта

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						27

4.3. Модуль распознавания лиц

Данный модуль состоит из трех несвязных частей – распознавания лиц в видеопотоке, распознавания лиц со статичного изображения, детектирования лиц со статичного фото.

4.3.1. Детектирование лиц со статичного изображения

Для разработки функции детектирования лиц использовался алгоритм, основанный на методе Виолы-Джонса. Данный метод основывается на примитивах Хаара, представляющих собой разбивку заданной прямоугольной области на наборы разнотипных прямоугольных подобластей:

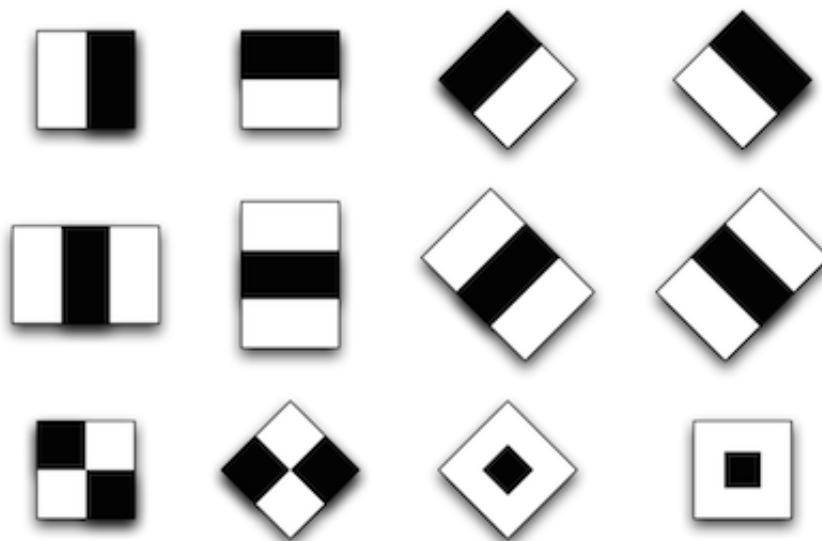


Рисунок 20 – примитивы Хаара

Для того, чтобы найти лицо, нужно выделить его основные компоненты, такие как нос, глаза, лоб, губы. Для этого существуют специальные шаблоны (примитивы) Хаара:

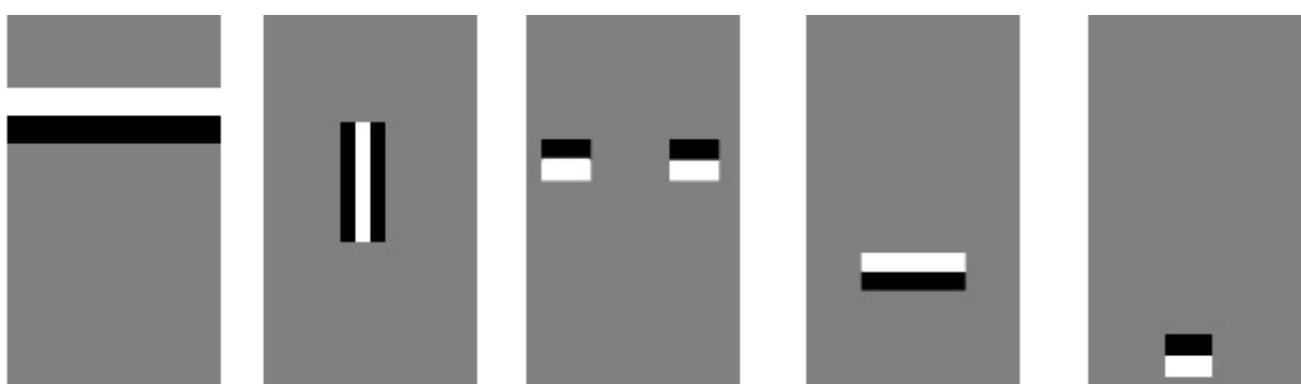


Рисунок 21 – примитивы Хаара для: лба, носа, глаз, губ, подбородка (слева направо)

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

Код специальности

Лист

28

Для каждого из этих шаблонов, высчитывается разность между яркостью белой и чёрной областей. Это значение сравнивается с эталоном и принимается решение о том, найдено лицо на фото или нет. В этом заключается метод Виолы-Джонса, который успешно используется в сфере компьютерного зрения.

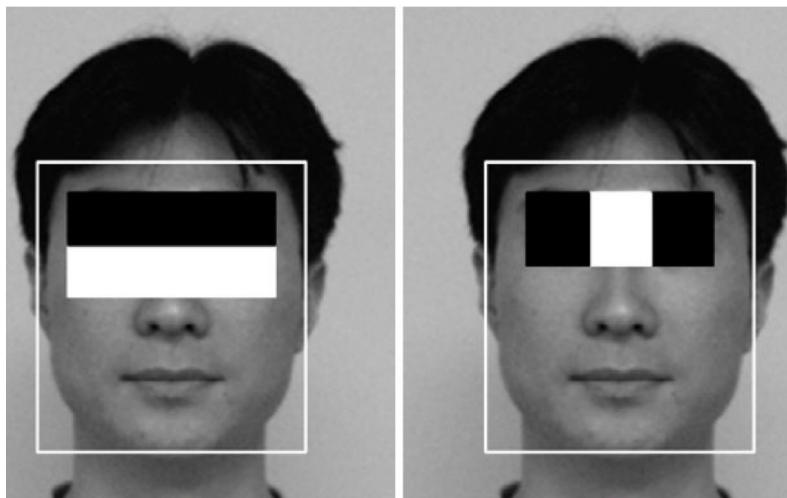


Рисунок 22 – пример использования шаблонов Хаара

Библиотека распознавания OpenCV имеет свою имплементацию алгоритма Виолы-Джонса, использование заключается в написании следующего кода:

```
1 import cv2 as cv
2
3 face_cascade = cv.CascadeClassifier('haarcascade.xml')
4 faces = face_cascade.detectMultiScale(#COLOR, 3, 5)
```

Создание класса *CascadeClassifier* происходит на основе файла с шаблонами (примитивами) в формате XML. Данный класс содержит функцию *detectMultiScale()*, которая выполняет предобработку изображения – преобразует его в черно-белый цвет и выполняет базовую цветокоррекцию, что позволяет увеличить эффективность детектирования. После предобработки функция применяет метод Виолы-Джонса, и возвращает результат в виде массива с координатами лица на изображении.

4.3.2. Распознавание лиц со статического изображения

Распознавание лица - это идентификация личности человека с заданного изображения на основе имеющихся баз данных. Технология распознавания требует минимум два этапа:

- ”обучение” системы на конкретные лица или массив лиц;

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						29

- распознавание на основе ”обучающих” данных.

Под термином ”обучение” в данном случае подразумевается преобразование изображения в вид, удобный и эффективный для сравнения. Для решения данной задачи использовался алгоритм Local Binary Patterns (LBP). Данный алгоритм состоит из следующих этапов:

- разделить изображение на части;
- в каждой части интенсивность цвета центрального пикселя сравнивается с интенсивностью соседних пикселей;
- если значение центрального больше соседнего, соседний пиксель заменяется на 0, иначе на 1. В результате получается некоторое число (см. Рисунок 23);
- на основе полученного числа строится гистограмма;
- гистограммы всех частей объединяются в один вектор, характеризующий изображение в целом. Такой числовой вектор можно использовать для сравнения и хранить в базе данных.

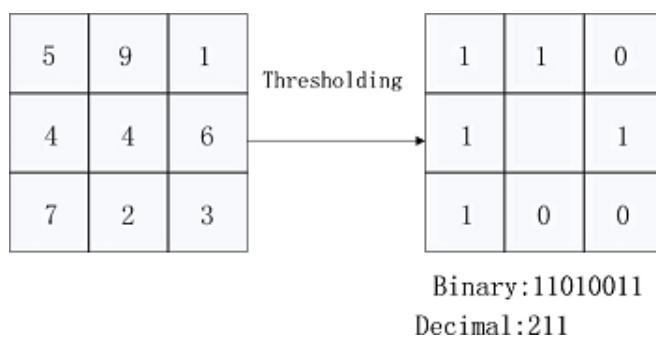


Рисунок 23 – Вычисление числа части изображения

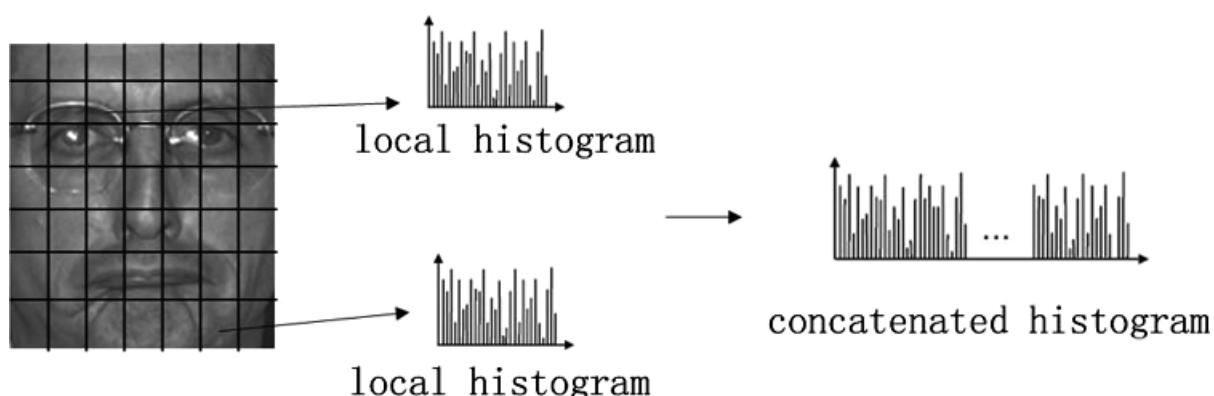


Рисунок 24 – Вычисление общей гистограммы и построение вектора

В библиотеке Face Recognition использовать алгоритм Local Binary Patterns можно следующим образом:

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						30

```

1 import cv2 as cv
2 import face_recognition
3
4 # Вычисление вектора полученного изображения
5 # IMAGE – изображение, COORDINATES – координаты найденного лица
6 encodings = face_recognition.face_encodings(IMAGE, COORDINATES)
7
8 # Сравнение полученного вектора с имеющимся в базе
9 # В случае распознавания функция вернёт значение True, иначе False
10 matches = face_recognition.compare_faces(knownEncodings, encoding)

```

4.3.3. Тренировка распознавания системы

Модуль распознавания требует наличия базы данных с лицами. Для каждого человека, которому должен быть разрешён доступ, нужно создать запись в базе данных с числовым вектором тренировочного фото, для дальнейшего сравнивания с кадрами из видеопотока.

Для решения задачи использовалась библиотека pickle – это база данных, основанная на файлах, и хранящая информацию в виде ”ключ - значение”. Подобный формат хранения удобно использовать для тренировки системы распознавания – достаточно получить числовой вектор лица, и сохранить в базе данных в виде ”имя - вектор”.

```

1 import pickle
2
3 # Чтение базы данных из файла
4 # PATH – путь хранения файла
5 data = pickle.loads(open(PATH, "rb").read())
6
7 # Получение из базы данных массива имён и массива векторов
8 knownEncodings = data["encodings"]
9 knownNames = data["names"]
10
11 # Формирование словаря ассоциативный( массив)
12 # и сохранение в базу данных
13 data = {"encodings": knownEncodings, "names": knownNames}
14 f = open(PATH, "wb")
15 f.write(pickle.dumps(data)).close()

```

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

4.3.4. Распознавание лиц с видеопотока

Используя детектирование и распознавание лиц был разработан модуль, описываемый в данной главе. Алгоритм работы модуля заключается в использования видеопотока с камеры с помощью библиотеки picamera,

```
1 import picamera  
2 from picamera import VideoStream  
3  
4 stream = picamera.VideoStream(0)  
5 while True:  
6     # Получение кадров с камеры в бесконечном потоке  
7     (ret, frame) = stream.read()
```

и сравнивании изображения с потока с изображениями из базы данных.

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

Код специальности

Лист

32

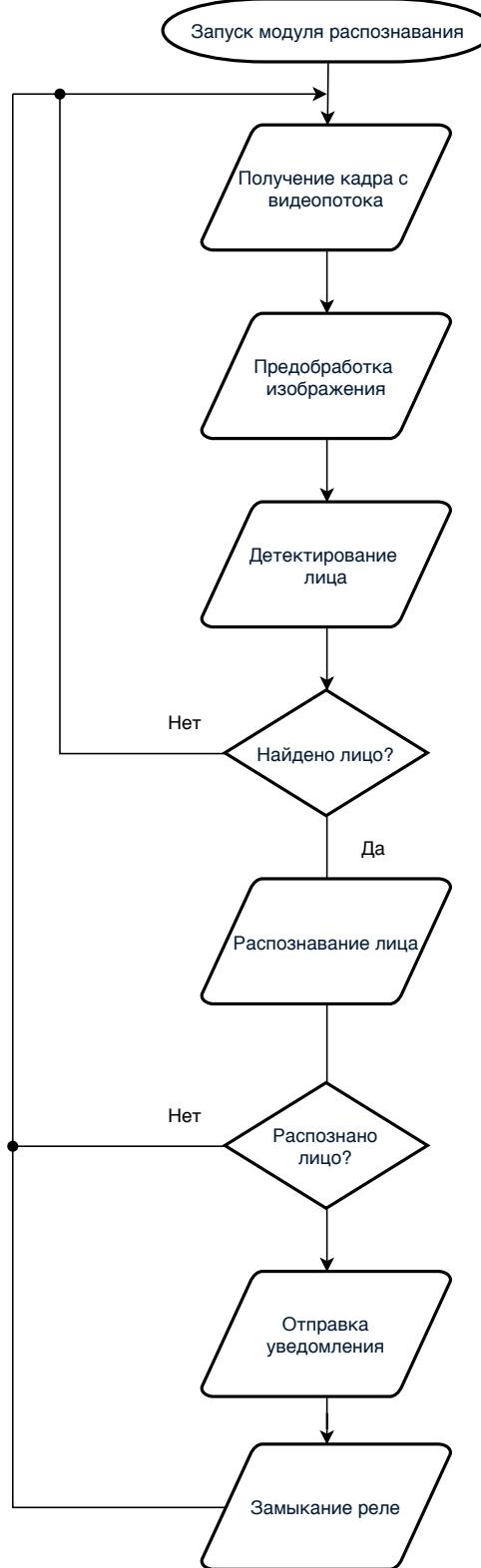


Рисунок 25 – Алгоритм работы модуля распознавания

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

4.4. Модуль Telegram-бот

Telegram-бот представляет в СКУД средство пользовательского управления работой системы. Бот связует две части – серверную и клиентскую с помощью серверов Telegram (см. Рис 26).

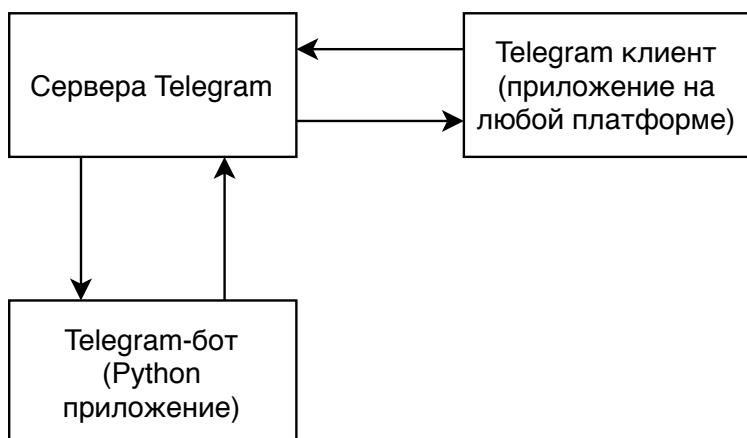


Рисунок 26 – Структура связи Telegram-бота и клиента

Функционал бота определяет:

- беспрерывное обновление данных;
- построение пользовательских меню;
- составление диалога с пользователем;

4.4.1. Получение и отправка данных Telegram-ботом

В дипломном проекте использовалась программная ”обёртка” над Telegram Bot API – библиотека pyTelegramBotAPI, выполняющая API запросы функцией Python.

Библиотека инкапсулирует все вызовы API в один класс – TeleBot. Для запуска бота требуется создать экземпляр класса Telebot (см. Рисунок 27)

```
1 from telebot import TeleBot  
2  
3 # TOKEN – уникальный код владельца бота, можно генерировать в  
4 # Father Bot, родительском боте.  
5 bot = TeleBot(TOKEN)
```

Для получения новых сообщений, требуется создать специальные функции с параметрами ожидаемых сообщений (тип, условие и т.п.):

```
1 # Функция для обработки сообщения с командой Статус ""  
2 @bot.message_handler(func=lambda message: message.text == "Статус")
```

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

```

3 | def cmd_status(msg):
4 |     # do something
5 |
6 | # Функция для обработки сообщения, содержащего
7 | # изображение
8 | @bot.message_handler(content_types=['photo'])
9 | def photo_msg(msg):
10|    # do something

```

Отправка новых сообщений выполняется функциями из класса TeleBot:

```

1 # Отправка сообщения
2 # CHAT_ID – идентификатор чата, RESPONSE – сообщение
3 bot.send_message(CHAT_ID, RESPONSE)
4
5 # Отправка изображения
6 bot.send_photo(CHAT_ID, photo)

```

4.4.2. Построение пользовательских меню

Управление ботом текстовыми командами неудобно для пользователя по причине долгого набирания команд, а также необходимости запоминания этих команд. Для избежания такого рода проблем были использованы встроенные в Telegram пользовательские меню. Подобный инструмент позволяет строить меню из кнопок, а по нажатию кнопки отправлять команду:

```

1 from telebot.types import InlineKeyboardButton, InlineKeyboardMarkup
2
3 keyboard = InlineKeyboardMarkup()
4 keyboard.row(InlineKeyboardButton(TEXT, COMMAND))
5
6 # Отправка сообщения с меню – кнопка COMMAND
7 bot.send_message(CHAT_ID, RESPONSE, reply_markup=keyboard)

```

4.4.3. Построение диалога с пользователем

Некоторые операции управления СКУД требуют больше одного сообщения от пользователя, а также нуждаются в уже полученных данных. Для решения проблемы бот был разработан на основе конечного автомата с использованием состояний.

При каждой команде, бот переходит в определенное состояние, и запрашивает новые данные. В конце графа каждого состояния бот сбрасывает его на

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						35

начальное. В ходе разработки Telegram-бот был реализован с использованием конечного автомата (Рис. 28).

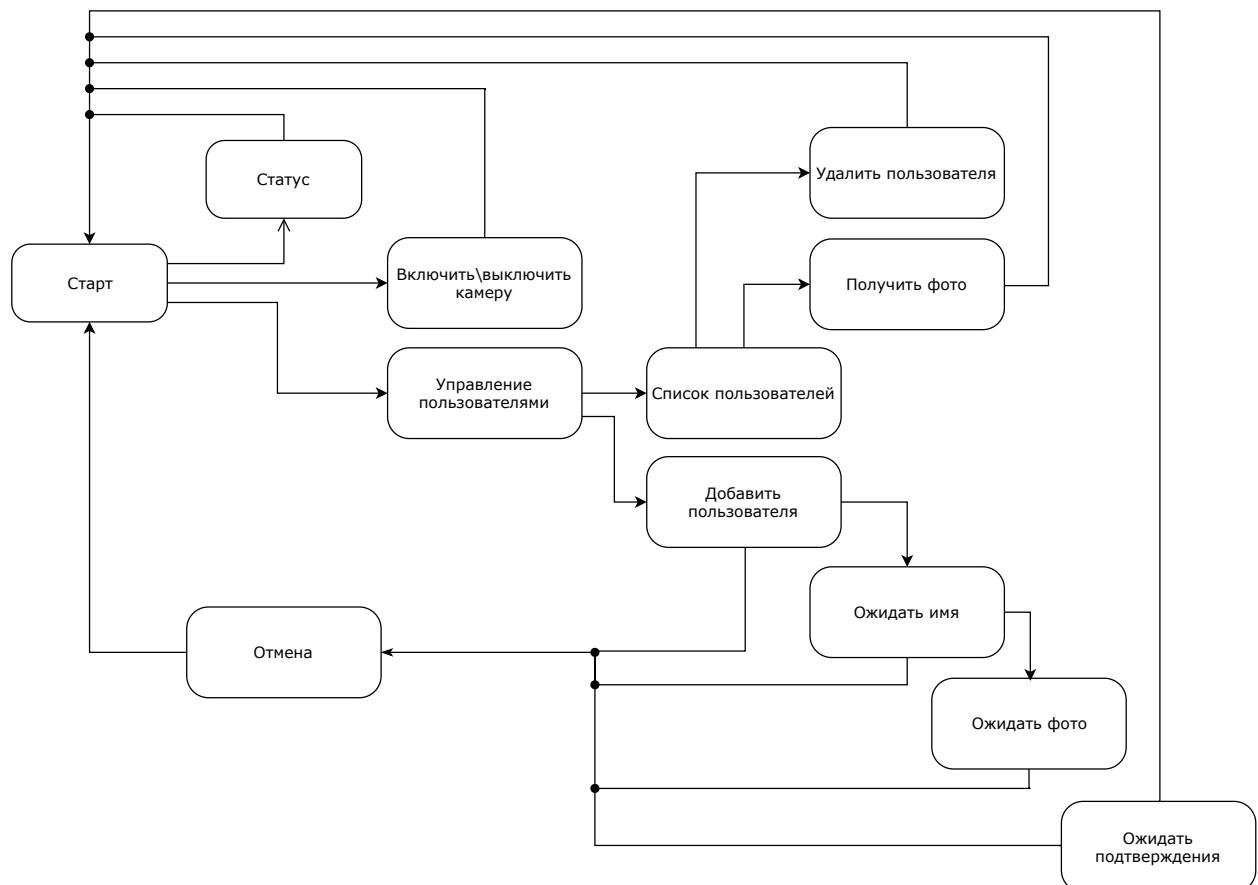


Рисунок 27 – Конечный автомат

5. Экономическая часть

- 6.1 расчет себестоимости
- 6.2 оценка эффективности внедрения

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						37

6. Охрана труда и экология

Нужно ли это делать?

- 7.1 вредные факторы
- 7.2 правила техники безопасности
- 7.3 противопожарная безопасность

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

Код специальности

Лист

38

Заключение

					Код специальности	Лист
						39
Изм.	Лист	№ докум.	Подп.	Дата		

Список литературы

Изм.	Лист	№ докум.	Подп.	Дата	Код специальности	Лист
						40