



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)

Институт опережающих технологий «Школа Икс»

Директор ИОТ «Школа Икс»
П.В. Герасин
подпись И.О. Фамилия
«_____» _____ марта 2024г.

КУРСОВОЙ ПРОЕКТ

Тема: _____ «Цифровая инфраструктура»

Дисциплина (модуль) _____ «Цифровая инфраструктура»
наименование учебной дисциплины (модуля)

Направление подготовки _____ 15.03.06 Мехатроника и робототехника
код и наименования направления подготовки

Направленность (профиль) _____

Обозначение курсового проекта _____ Группа ХР21

Обучающийся _____ Р. А. Лемешкин
подпись дата И.О.Фамилия

Курсовой проект защищен с оценкой _____

Руководитель проекта _____ К.Т.Н., доц. А.В. Авилов
подпись, дата должность И.О.Фамилия

Ростов-на-Дону
2024



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Институт опережающих технологий «Школа Икс»

ЗАДАНИЕ

на выполнение курсового проекта

Тема: _____ «Цифровая инфраструктура» _____

Дисциплина (модуль) _____ «Цифровая инфраструктура» _____
наименование учебной дисциплины (модуля)

Обучающийся _____ Лемешкин Руслан Андреевич _____
Фамилия Имя Отчество полностью

Обозначение курсового проекта _____ Группа ХР21

Исходные данные для выполнения курсового проекта:

Методические указания к выполнению курсового проекта.

Содержание курсового проекта:

Введение:

Целью работы является разработка системы «Умный дом», а также реализация в ней охранной системы видеонаблюдения. Помимо этого было необходимо отобразить работу системы на стенде и реализовать вывод данных с датчиков на сайт.

Наименование и краткое содержание разделов:

1. Анализ целевой аудитории проекта
2. Создание системы «Умный дом»
3. Система компьютерного зрения
4. Разработка web-сайта
5. Создание бота в Telegram

Заключение: Была разработана система «Умный дом», в которой данные с датчиков выводятся на сайт, также была реализована охранная система (данные с камеры видеонаблюдения отправляются в бот в Telegram). Работа системы отображена на стенде

Руководитель проекта _____ К.Т.Н., доц. _____ А.В. Авилов _____
подпись, дата должность И.О.Фамилия

Обучающийся _____ Р. А. Лемешкин _____
подпись дата И.О.Фамилия

Содержание

Содержание	1
1 Введение	2
2 Анализ целевой аудитории проекта.	3
3 Создание системы “Умный дом”	5
3.1 Внешний вид стенда Умный дом	5
3.2 Алгоритм работы системы «Умный дом»	6
3.3 Датчики и серверная часть системы	6
3.3.1 Используемые программы	6
3.3.2 Логика работы	8
3.3.3 Особенности реализации	11
4 Система компьютерного зрения	12
4.1 Выбор ИИ для проекта	12
4.2 Выбор технологического стека	13
4.3 Реализация	13
4.3.1 Структура программы:	14
5 Разработка web-сайта.	18
5.1 Анализ существующих web-страниц	18
5.2 Создание дизайна интерфейса	19
5.3 Разработка веб-страницы	22
6 Телеграмм бот	25
6.1 Используемые материалы для создания телеграмм бота	25
6.2 Функции телеграмм бота	26
7 Заключение	29
8 Перечень использованных информационных ресурсов	30

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	5.1	Анализ существующих web-страниц	18					
					5.2	Создание дизайна интерфейса	19					
					5.3	Разработка веб-страницы	22					
					6	Telegramм бот	25					
					6.1	Использованные материалы для создания telegramм бота	25					
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	6.2	Функции telegramм бота	26					
					7	Заключение	29					
					8	Перечень использованных информационных ресурсов	30					
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата								
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	Изм	Лист	№ докум.					
					Подп.	Дата	15.03.06.930000.000 ПЗ					
					Разраб.	Лемешкин			Умный дом Пояснительная записка	Лит.	Лист	Листов
					Пров.	Авилов А.В					1	30
					Н.контр.					ДГТУ ИОТ «Школа Икс»		
Утв.	Герасин П. В.											

1 Введение

Современные технологии систем умных домов предоставляют уникальные возможности для автоматизации и управления различными аспектами жилища, обеспечивая комфорт, безопасность и энергоэффективность. В данной работе был проведен анализ целевой аудитории проекта, с целью определения потребностей и предпочтений пользователей.

Наша задача была разработать систему «Умный дом». Система в виде стенда с датчиками, которые отображают её работу. Также, в эту систему входят камеры, которые отправляют информацию о безопасности в наш Телеграмм бот.

В рамках исследования, было изучено применение системы компьютерного зрения в проекте, включая роль и необходимость искусственного интеллекта, различные типы искусственного интеллекта, выбор языка программирования и библиотеки, а также этапы реализации данной технологии.

Далее разработка web-сайта с анализом существующих страниц, созданием дизайна интерфейса, frontend и backend разработкой. Создание бота в Telegram, включая использованные материалы, функционал и цели данного проекта. Соединением сайта с датчиками также было реализовано.

Каждый раздел представляет собой важный этап в разработке и реализации системы умного дома, направленный на обеспечение удобства, эффективности и современных технологических решений для пользователей.

Изм.	Лист	№ докум.	Подп.	Дата	Изм.	Лист	15.03.06.930000.000 ПЗ	2																		
									Изм.	Лист	№ докум.	Подп.	Дата	Изм.	Лист	15.03.06.930000.000 ПЗ	2									
																		Изм.	Лист	№ докум.	Подп.	Дата	Изм.	Лист	15.03.06.930000.000 ПЗ	2

2 Анализ целевой аудитории проекта.

Для определения нашей целевой аудитории мы провели анализ рынка.

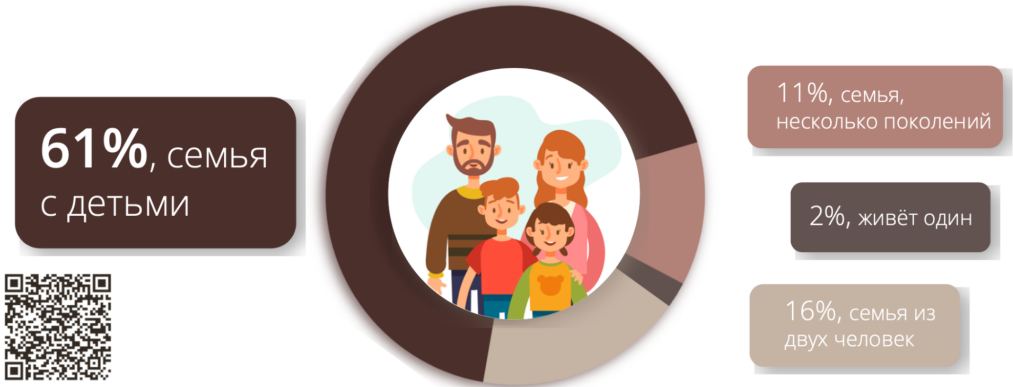
Целевая аудитория

Наша целевая аудитория: семьи с детьми

Доход: выше среднего (от 200 т.р)

Возраст: 25-35 лет

Работа: удаленно



Пол: мужчины (68%), женщины (32%)

Возрастные группы: от 25 до 34 лет (40%), от 35 до 44 лет (28%)

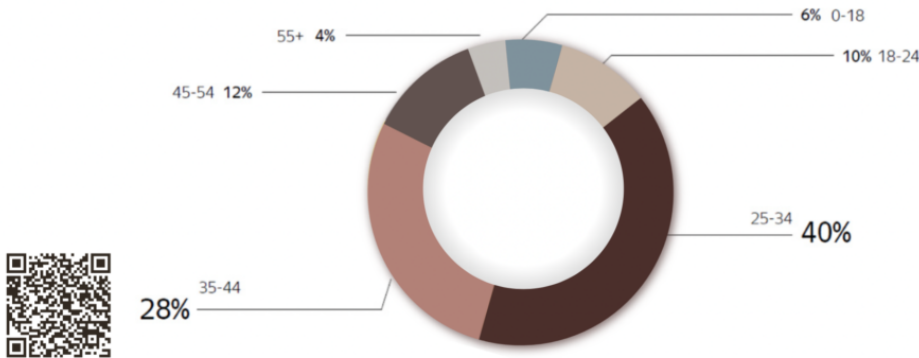


Рисунок 2.1 — статистические данные для анализа целевой аудитории.

Основываясь на статистических данных можно сделать вывод о том, что наибольшей группой покупателей системы умного дома являются мужчины в возрасте от 25 до 34 лет [1]. Также благодаря статистическим данным мы выявили, что 61% покупателей умных домов являются семьи с детьми[2]. Помимо этого, мы прописали, что жители наших умных коттеджей должны работать удалённо. Это связано с тем, что коттеджные поселки находятся вдали от центра города. Люди, работающие в офисах предпочитают выбирать жилье, которое находится как можно ближе к их работе, следовательно им будет гораздо комфортнее жить в жилых комплексах в квартире. Но, если у покупателей

Инт. № подл.	Подп. и дата
Взам. инв. №	Инт. № дубл.
Подп. и дата	
Инт. № подл.	

Изм	Лист	№ докум.	Подп.	Дата	15.03.06.930000.000 ПЗ	Лист
						3

коттеджей нет привязки к месту (работаю удалённо), им нет смысла стремиться купить дом ближе к центру.

Помимо этого, при анализе рынка мы выявили основные запросы нашей целевой аудитории:

- а) Безопасность. Благодаря системе видеонаблюдения с распознаванием силуэта человека, а также датчику утечки газа, семья может быть спокойна за безопасность своих близких.
- б) Климат-контроль. Благодаря датчикам температуры и влажности пользователь сможет отслеживать данные с них на сайте.
- в) Удобство и комфорт. Коттеджный поселок располагается в удобном для проживания месте. Также есть несколько планов домов, которые смогут удовлетворить потребности разных семей.
- г) Дистанционное управление. Владельцы смогут управлять умным домом через Telegram-bot, а также отслеживать данные с датчиков через сайт.
- д) Помощь по дому. Умный робот будет помогать убирать игрушки за детьми.

Изм.	Лист	№ докум.	Подп.	Дата	15.03.06.930000.000 ПЗ	Изм.	Лист	№ докум.	Подп.	Дата

3.2 Алгоритм работы системы «Умный дом»

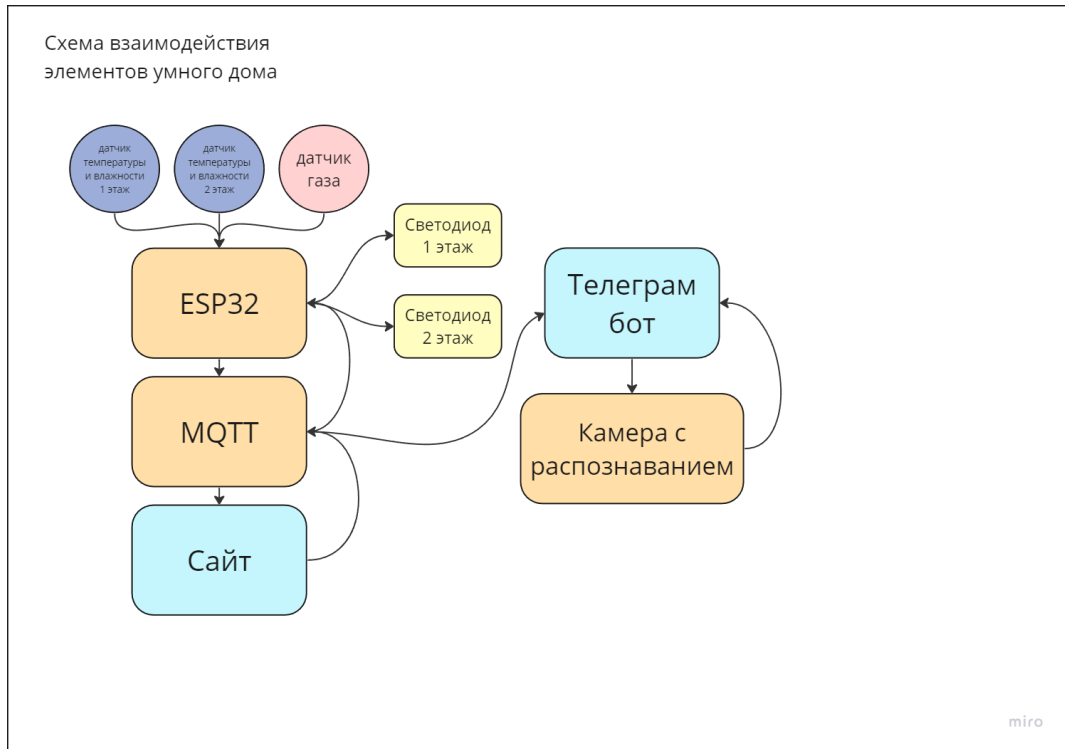


Рисунок 3.2 — алгоритм работы системы «Умный дом».

На рисунке 3.2 мы видим, что микроконтроллер получает данные с двух датчиков температуры и влажности DHT11 и управляет двумя трехцветными светодиодами. При этом ESP32 обменивается данными с mqtt сервером. С mqtt сервера данные поступают с датчиков поступают на сайт и телеграм бота. С сайта производится включение и выключения светодиодов, имитируя свет в доме. На телеграм бота поступают данные с датчиков температуры и влажности и газа, обратно бот не отдает никакие данные. С телеграм бота пользователь может узнать данные с датчиков и взаимодействовать с камерой которая распознает силуэты людей.

3.3 Датчики и серверная часть системы

3.3.1 Используемые программы

Mosquitto — локальный mqtt брокер, позволяющий сайту, телеграм-боту и стенду на esp32 обмениваться данными.

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата

Thonny IDE — интегрированная среда разработки для языка micropython, необходимая для написания и отладки кода микроконтроллера, а также последующей прошивки устройства.

nano — консольный текстовый редактор, необходимый для исправления конфигурационных файлов и подготовки системы для установки прочих программ.

emerge — интерфейс командной строки для portage, пакетного менеджера ОС gentoo по-умолчанию, необходимый для установки программ и библиотек, а также для управления их зависимостями.

nmcli — интерфейс командной строки для Network Manager, программы для управления сетевыми подключениями в ОС, основанных на GNU/Linux, необходимый для переключения в зависимости от ситуации между глобальной сетью Интернет и локальной сетью, которую создавала esp32.

ip — консольная утилита, позволяющая узнать свой локальный IP, необходимый для подключения микроконтроллера к mqtt брокеру.

lsof — консольная утилита, позволяющая узнать процессы в системе, занимающие определенные сетевые порты. Данная информация была необходима для исследования системы и управления различными версиями сайта, запущенными параллельно для сравнительного анализа и разработки наилучшей комбинации различных решений.

rm — консольная утилита, необходимая для удаления ошибочно созданных и мешающих файлов.

mkdir — консольная утилита, необходимая для создания новых директорий для проекта.

micropython — консольный интерфейс для REPL, среды исполнения языка micropython.

esptool — консольная утилита для прошивки микроконтроллеров esp.

3.3.2 Логика работы

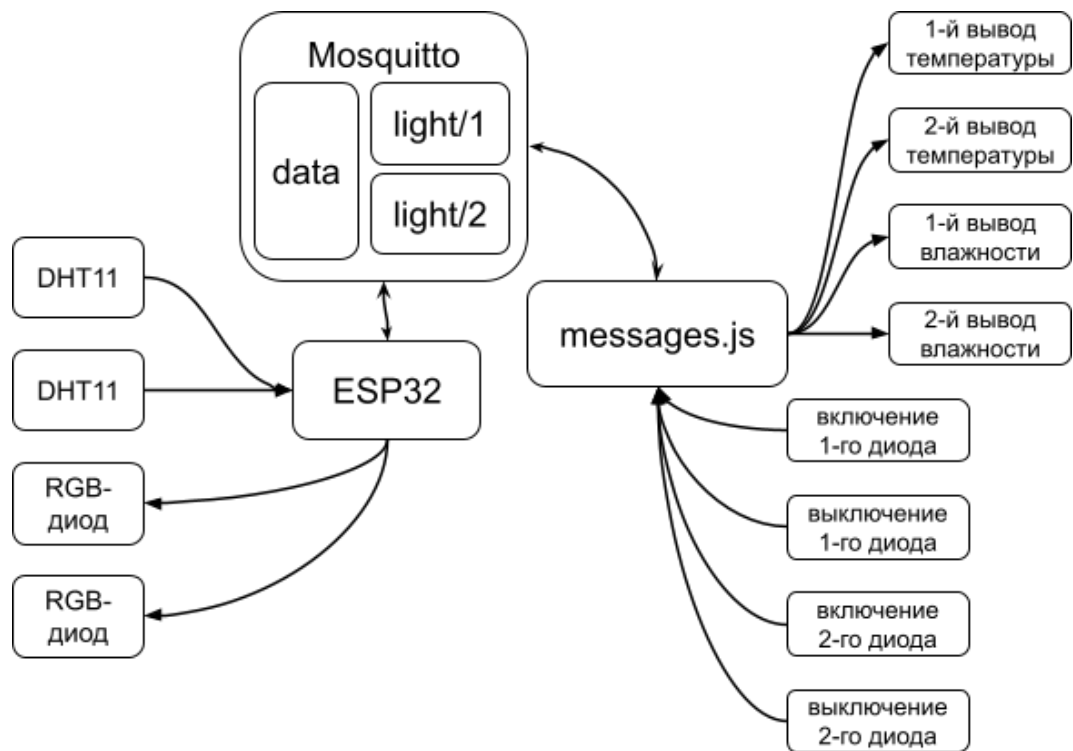


Рисунок 3.3 — Схема логики работы

На схеме видно, что микроконтроллер получает данные с двух датчиков температуры и влажности DHT11 и управляет двумя трехцветными светодиодами. При этом ESP32 обменивается данными с mqtt сервером Mosquitto, отправляя данные с датчиков в топик data и читая из топиков light/1 и light/2.

Сайт представлен здесь файлом messages.js и несколькими элементами верстки, связанными с ним. Подробнее его устройство будет рассмотрено в разделе Backend сайта.

Рассмотрим подробнее логику работы ESP32. Как было указано выше, для программирования микроконтроллера использовались IDE thonny и язык программирования micropython (далее ЯП или micropy). В micropy большая часть функционала вынесена в библиотеки, поэтому их пришлось активно использовать при программировании. Так для взаимодействия и управления светодиодами использовалась функция Pin библиотеки machine, для взаимодействия с датчиками использовалась библиотека dht. Также использовались библиотеки network для сети, umqtt.simple для передачи данных и time для выставления

Инв. № подл.	Подш. и дата	Взам. инв. №	Инв. № дубл.	Подш. и дата	<div>На схеме видно, что микроконтроллер получает данные с двух датчиков температуры и влажности DHT11 и управляет двумя трехцветными светодиодами. При этом ESP32 обменивается данными с mqtt сервером Mosquitto, отправляя данные с датчиков в топик data и читая из топиков light/1 и light/2.</div> <div>Сайт представлен здесь файлом messages.js и несколькими элементами верстки, связанными с ним. Подробнее его устройство будет рассмотрено в разделе Backend сайта.</div> <div>Рассмотрим подробнее логику работы ESP32. Как было указано выше, для программирования микроконтроллера использовались IDE thonny и язык программирования micropython (далее ЯП или micropу). В micropу большая часть функционала вынесена в библиотеки, поэтому их пришлось активно использовать при программировании. Так для взаимодействия и управления светодиодами использовалась функция Pin библиотеки machine, для взаимодействия с датчиками использовалась библиотека dht. Также использовались библиотеки network для сети, umqtt.simple для передачи данных и time для выставления</div>	
Изм.	Лист	№ докум.	Подш.	Дата	15.03.06.930000.000 ПЗ	Лист
						8

задержки между снятиями показателей или запросом новых статусов для светодиодов. На рисунке 6 представлены импорты необходимых компонентов.

```
1 from time import sleep
2 from machine import Pin
3 from network import WLAN, AP_IF, STA_IF
4 from umqtt.simple import MQTTClient
5 from dht import DHT11
```

Рисунок 3.4 — Импорты необходимых компонентов

```
7 def sub(topic, msg):
8     global light_changer
9     topic, msg = str(topic), str(msg)
10    light_changer = [topic[-1], msg[-1]] if 'light' in topic else None
11    print(f'received message {msg} on topic {topic}')
```

Рисунок 3.5 — функция для обработки входящих сообщений

На рисунке 3.4 показана функция для обработки входящих сообщений. Здесь используется редактирование глобальной переменной light_changer из-за особенностей реализации umqtt.simple. В неё помещаются последние символы из переменных topic & msg, которые должны являться целыми числами, которые обозначают номер светодиода (1 или 2) и его состояние (0 – выключен и 1 – включён). Также выводится отладочное сообщение об успешности получения сообщения.

```
13 wlan = WLAN(AP_IF) if input('input mode: ') == 'local' else WLAN(STA_IF)
14 wlan.active(True)
15 try:
16     wlan.config(ssid="Mesto-AP",
17                 security=3,
18                 key="Mesto123")
19 except OSError:
20     wlan.connect('ASUS-SVK ', '88888888')
```

Рисунок 3.6 — подключение к сети

На рисунке 3.6 показано подключение к сети. Здесь создаётся экземпляр класса WLAN в раздающем или принимающем варианте соответственно, а затем активируется. Далее использован механизм обработки исключений, который пытается настроить собственную сеть, а при неудаче подключается к существующей сети. Данный код использует особенность создаваемого экземпляра, который может выполнить только одно из указанных действий, а при попытке выполнения другого возвращает исключение.

Изн. № подл.	Подп. и дата	Взам. инв. №	Изн. № дубл.	Подп. и дата

```

22 client = MQTTClient("esp32-home", "192.168.4.2", 1883, "Mesto-MQTT", "Mesto123")
23 client.set_callback(sub)
24 client.connect()
25 client.subscribe("light/1")
26 client.subscribe("light/2")
27 print('connected')

```

Рисунок 3.7 — подключение к mqtt-брокеру

Здесь показано подключение к mqtt-брокеру. Важно, что уведомление о подключении приходит только после подключения к Mosquitto, ведь до него нет полезного соединения. Также этот код осуществляет подписку на топики.

Данный код просто объявляет переменные для обмена статусом светодиодов, управления светодиодами и датчиками соответственно.

```

29 light_changer = None
30 lights = ["blank", Pin(12, Pin.OUT), Pin(14, Pin.OUT)]
31 odht1 = DHT11(Pin(25))
32 odht2 = DHT11(Pin(32))

```

Рисунок 3.8 — объявление важных переменных

```

34 while True:
35     client.check_msg()
36     if light_changer is not None:
37         lights[light_changer[0]].value(light_changer[1])
38         light_changer = None
39     odht1.measure()
40     odht2.measure()
41     client.publish("data", str(odht1.temperature())+' '
42                             +str(odht2.temperature())+' '
43                             +str(odht1.humidity())+' '
44                             +str(odht2.humidity()))
45     sleep(5)

```

Рисунок 3.9 — тело рабочего цикла программы

Теперь представлено тело рабочего цикла программы. Здесь проверяется наличие новых сообщений во всех топиках, на которые есть подписка, а затем назначается новое значение включённости светодиодам. Далее измеряются температура и влажность, которые собиются в одну строку для удобства передачи и публикуются в топик data. После этого всего идёт ожидание в 5 секунд и цикл повторяется.

Инд. № подл.	Подп. и дата	Инд. № дубл.	Подп. и дата
Взам. инв. №			
Инд. № подл.	Подп. и дата	Инд. № дубл.	Подп. и дата

3.3.3 Особенности реализации

В данном разделе описывается работа над запуском сервера и микроконтроллера, а также наладкой связи между ними.

Как операционная система для сервера Mosquitto была выбрана Gentoo Linux. Для установки Mosquitto использовался emerge, при этом для снятия маскировки некоторых пакетов пришлось править конфигурационные файлы через терминал в силу особенностей установленной системы с помощью текстового редактора nano и средств управления файловой системой rm & mkdir.

Также предварительной прошивки и настройки требовал микроконтроллер. Для его прошивки потребовалась программа esptool, которая также есть в репозиториях portage и сам файл прошивки с официального сайта. Инструкция использована с сайта micropython.org. Также для продуктивной разработки под ESP32 используется thonny IDE, которая устанавливается через pip, но при попытке установить в общее пространство системы, но процесс был прерван системой из-за ошибки, заключающейся в том, что общее пространство уже контролируется пакетным менеджером portage. Данное ограничение получилось обойти через создание виртуального окружения с помощью venv.

Следующим шагом стала настройка сети. Было предусмотрено 2 варианта: работа в локальной сети, создаваемой ESP32 и работа в сети с выходом в интернет, создаваемой роутером. Переключение между вариантами соединения производится посредством введения команды local в момент установки соединения либо пропуска поля ввода. Для обоих вариантов было необходимо узнать IP адрес сервера, что было сделано командой ip. Далее адрес был использован в программе микроконтроллера и подключение успешно установлено. Особенности каждого из вариантов подключения со стороны микроконтроллера можно увидеть выше по тексту. Со стороны сервера подключение всегда производится командой nmcli с указанием SSID и пароля сети. Также nmcli имеет функцию хранения паролей, поэтому вводить их требуется только при первом подключении.

При настроенной сети связь с mqtt-брокером уже не вызвала трудностей и данные были успешно переданы с датчиков на сервер.

Ив. № подл.	Подп. и дата	Взам. инв. №	Ив. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата	Ив. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата
					<p>нашего умного дома обеспечивает надежную защиту и моментальное реагирование на потенциальные угрозы безопасности. Благодаря ИИ, владелец умного дома может быть уверен в надежной защите своей территории и оперативном реагировании на любые возможные инциденты.</p> <p>Таким образом, использование Искусственного Интеллекта в нашем проекте умного дома не только повышает уровень безопасности, но и обеспечивает комфорт и уверенность владельцу, что его территория надежно защищена.</p> <p>В мире искусственного интеллекта существует несколько типов, включая слабый и сильный ИИ. Слабый ИИ ограничен в своих возможностях и решает задачи в рамках конкретной области, например, системы распознавания речи или обработки изображений. Сильный ИИ, наоборот, обладает способностью обучаться и принимать решения в различных областях, как человек. Кроме того, существуют экспертные системы, которые используют знания экспертов в виде правил для принятия решений, и нейронные сети, которые моделируют работу человеческого мозга.</p>				
					<p>15.03.06.930000.000 ПЗ</p>				
					<p>Лист 12</p>				

Для нашего проекта умного дома подходит использование слабого ИИ, а именно системы компьютерного зрения на основе нейронных сетей. Это позволяет нам обнаруживать и классифицировать объекты на видео, такие как люди, транспортные средства и животные, с высокой точностью.

4.2 Выбор технологического стека

Для разработки программного обеспечения нашего проекта мы выбрали язык программирования Python, который получил широкую популярность и имеет богатую экосистему инструментов и библиотек для машинного обучения и обработки данных. Python обеспечивает гибкость и простоту в разработке, что позволяет создавать эффективные и масштабируемые приложения.

Для реализации системы компьютерного зрения мы выбрали библиотеку OpenCV, которая предоставляет множество функций для работы с изображениями и видео. OpenCV обеспечивает высокую производительность и гибкость в обработке видео и изображений. В частности, мы использовали нейросеть YOLOv8, которая является одной из наиболее популярных и эффективных нейросетей для обнаружения объектов на видео. YOLOv8 обеспечивает высокую точность и скорость обнаружения объектов, что позволяет использовать ее в различных приложениях, включая систему компьютерного зрения для нашего проекта умного дома.

4.3 Реализация

В нашем проекте мы использовали встроенную камеру от ноутбука в качестве источника видеопотока для системы компьютерного зрения. Для обработки видеоданных и обнаружения людей мы запустили нейросеть YOLOv8 на том же ноутбуке. Этот выбор позволил нам минимизировать затраты на оборудование и увеличить скорость быстрогодействия системы.

Мы настроили нейросеть YOLOv8 для обнаружения силуэтов людей на видеопотоке с камеры. Это включало в себя выбор соответствующей модели нейросети, ее обучение на обучающем наборе данных и настройку параметров для достижения оптимальной производительности.

На ноутбуке была запущена охранная программа, которая интегрировала работу камеры, нейросети YOLOv8 и телеграмм бота. Программа непрерывно

Инд. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

мониторила видеопоток с камеры, передавала его на обработку нейросети для обнаружения людей и отправляла уведомления через телеграмм бота в случае обнаружения посторонних лиц на территории дома.

Алгоритм работы охранной системы:

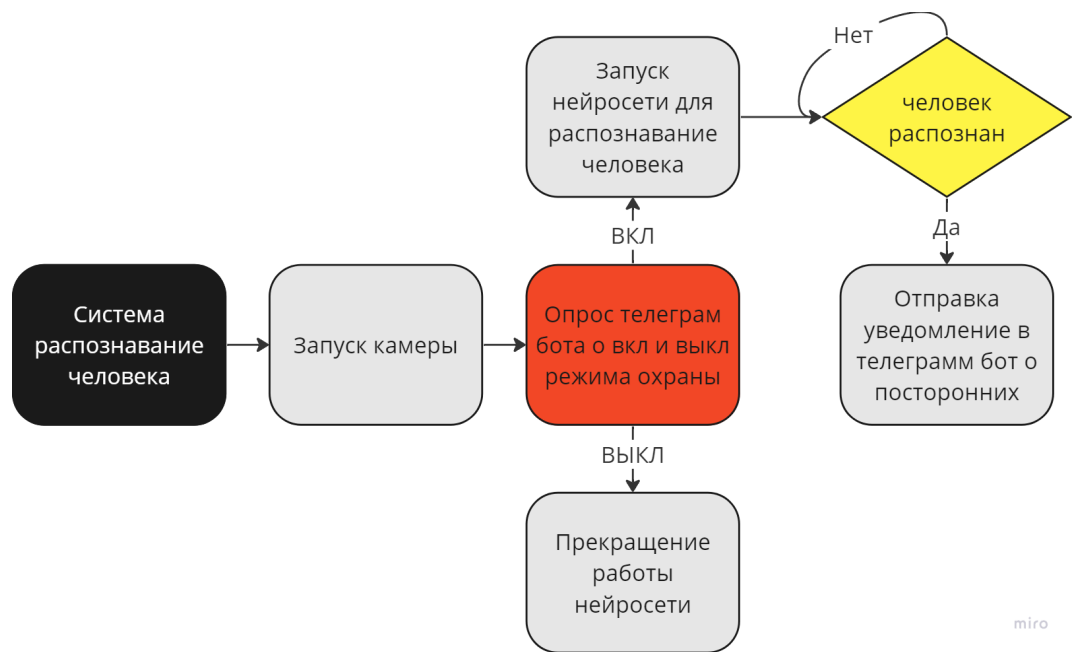


Рисунок 4.1 — алгоритм работы охранной системы.

4.3.1 Структура программы:

Программа охранной системы состоит из двух основных Python файлов:

Основной файл обработки видеопотока:

Этот файл отвечает за обработку видеопотока с камеры и обнаружение нарушений безопасности на придомовой территории. Он включает в себя инициализацию камеры, обработку видеопотока с использованием модели YOLOv8 для обнаружения людей, а также отправку уведомлений через Телеграм бота в случае обнаружения нарушения.

Файл с Телеграм ботом:

Этот файл содержит код Телеграм бота, который обрабатывает команды о включении и выключении охранной системы. Бот принимает команды от пользователя через интерфейс Телеграм мессенджера и управляет работой охранной системы, например, включая или отключая отправку уведомлений о проникновении.

Взаимодействие между файлами:

Для согласованной работы между основным файлом обработки видеопотока и

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата
Изм	Лист	№ докум.	Подп.	Дата

файлом с Телеграм ботом используется текстовый файл "send_photos.txt". Этот файл служит для передачи информации о необходимости отправки фотографий в Телеграм при обнаружении нарушений безопасности.

Телеграм бот обрабатывает команды пользователя и записывает соответствующие значения в текстовый файл. При получении команды "/photocheck" бот изменяет состояние переменной 'send_ph' и записывает соответствующее значение ("отправлять фото" или "не отправлять фото") в файл "send_photos.txt".

```
@bot.message_handler(commands=["photocheck"])
def start_com(message):
    print("photocheck")
    global send_ph
    send_ph = not send_ph
    if send_ph:
        with open("send_photos.txt", "w") as file:
            file.write("отправлять фото")
        bot.send_message(message, text="камера включена")
    else:
        with open("send_photos.txt", "w") as file:
            file.write("не отправлять фото")
        bot.send_message(message, text="камера выключена")
```

Рисунок 4.2 — обработка команды "/photocheck".

В основном цикле программы осуществляется чтение значения из текстового файла "send_photos.txt" для определения, нужно ли отправлять фотографии в Телеграм бота при обнаружении нарушений. Если значение в файле указывает на необходимость отправки фотографий и на кадре обнаружены нарушения безопасности, то программа осуществляет отправку фотографии через Телеграм бота.

Главный цикл программы продолжается до момента, когда пользователь не прерывает его. В этом цикле вызывается функция 'ai()' для обработки каждого кадра с камеры. В зависимости от настроек, указанных в файле "send_photos.txt", и наличия обнаруженных нарушений, может быть отправлено уведомление с фотографией в Телеграм.

Функция отправки фотографии в Телеграм:

Инд. № подл.	Подп. и дата	Взам. инв. №	Инд. № дубл.	Подп. и дата
Изм	Лист	№ докум.	Подп.	Дата

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
def ai():
    ret, img = capture.read()
    # Ищем на кадре человека (classes=0)
    results = model.predict(source=img, classes=0, conf=0.65, verbose=False)
    # Количество людей на видео
    num_people = results[0].boxes.shape[0]
    print("People = ", num_people)
    plot_res = results[0].plot(labels=True)
    cv2.imshow( winname: 'WebCam', plot_res)
    return [num_people,img]
```

Рисунок 4.5 — функция обработки видеопотока с камеры.

Основной цикл программы:
Главная функция:

```
if __name__ == "__main__":
    # Открыть видеопоток
    capture = cv2.VideoCapture(0)
    # Загрузить модель YOLO
    model = YOLO("yolov8n.pt")
    main()
```

Рисунок 4.6 — главная функции.

Чтобы обеспечить гибкость и модульность программы, в конце скрипта добавлен блок условия if __name__ == "__main__":. Этот блок позволяет запускать код только в случае, если файл выполняется непосредственно, а не импортируется как модуль в другой скрипт.

5 Разработка web-сайта.

Перед тем, как начать делать сайт, мы решили определить потребности нашей целевой аудитории.

В первую очередь веб-страница должна быть доступной для всех людей. Это включает в себя использование семантического HTML для правильного структурирования контента, а также обеспечение возможности навигации с клавиатуры.

Помимо этого интерфейс должен быть понятным и лёгким для восприятия, предоставляя пользователю простые инструкции для управления. Также управление сайтом должно быть логичным и последовательным для того, чтобы пользователь легко ориентировался и понимал, что происходит на странице. Кроме того, элементы дизайна и функциональности веб-страницы должны быть согласованы и создавать цельное впечатление, обеспечивая удобство использования.

5.1 Анализ существующих web-страниц

После анализа потребностей нашей целевой аудитории мы решили проанализировать сайты, которые уже существуют.

- а) Умный дом с Алисой [1]3 Дизайн и визуальное оформление страницы очень гармонично смотрятся, все расположено понятно для пользователя. Шрифт и композиция соответствуют целевой аудитории. Навигация и удобство использования выделяются на фоне других страниц, так как основные разделы размещены в привычных местах, что делает интерфейс более понятным. Сайт также адаптируется под различные устройства. Особенно привлекательными на сайте являются функциональность и интерактивность, такие как формы обратной связи, возможность комментирования, интерактивные элементы и другие функции. Интерфейс устраивает пользователей.
- б) Умный дом Korolab4 Korolab имеет удобный интерфейс, поскольку все структурировано по этажам. Дизайн и оформление простые и понятные. Шрифт и композиция не выделяются, все размещено стратегически. Однако сайт плохо адаптируется под разные устройства, поскольку в мобильном

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата	лизировать сайты, которые уже существуют.					
					а) Умный дом с Алисой [1]3 Дизайн и визуальное оформление страницы очень гармонично смотрятся, все расположено понятно для пользователя. Шрифт и композиция соответствуют целевой аудитории. Навигация и удобство использования выделяются на фоне других страниц, так как основные разделы размещены в привычных местах, что делает интерфейс более понятным. Сайт также адаптируется под различные устройства. Особенно привлекательными на сайте являются функциональность и интерактивность, такие как формы обратной связи, возможность комментирования, интерактивные элементы и другие функции. Интерфейс устраивает пользователей.					
					б) Умный дом Korolab4 Korolab имеет удобный интерфейс, поскольку все структурировано по этажам. Дизайн и оформление простые и понятные. Шрифт и композиция не выделяются, все размещено стратегически. Однако сайт плохо адаптируется под разные устройства, поскольку в мобильном					
Изм	Лист	№ докум.	Подп.	Дата	15.03.06.930000.000 ПЗ					Лист
										18

приложении не выглядит также эстетично, как на веб-странице. Отзывы пользователей положительные относительно интерфейса.

5.2 Создание дизайна интерфейса

Так как нашей целевой аудиторией являются молодые семьи с детьми (возраст родителей от 25 до 35 лет), наша веб-страница должна быть простой, удобной в использовании и понятной для пользователей. Также на сайте должны отображаться актуальные данные с датчиков в доме. Помимо этого, для удобства пользователей мы решили внедрить в сайт управление освещением.

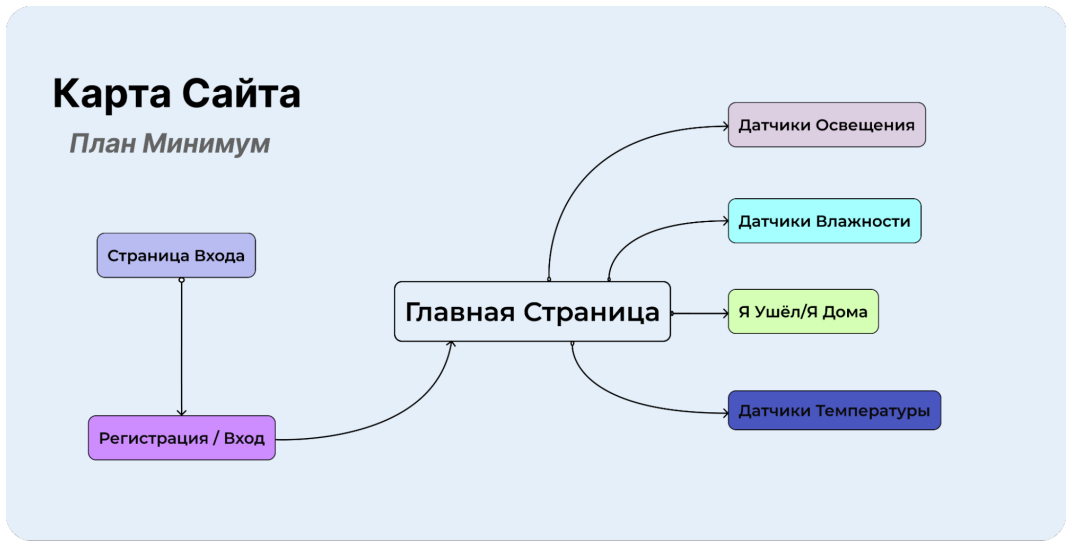


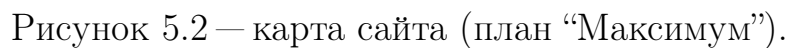
Рисунок 5.1 — карта сайта (план “Минимум”).

Для понимания того, как будет работать веб-страница мы решили сделать карты сайта. Оценивая наши возможности мы создали план минимум (который мы точно должны сделать), а также план максимум (который было бы хорошо сделать, если это получится).

Изначально мы планировали, чтобы пользователь входил на сайт при помощи пароля, но, после того, как мы решили сделать бота в Telegram, где пользователь будет входить в управление своим домом при помощи своего ID, мы решили пока что это не реализовывать. Также в плане минимум мы решили реализовывать одностраничный сайт, на котором можно просматривать информацию с датчиков, а также регулировать освещённость на первом и втором этаже. Помимо этого, в начале мы думали, включать и отключать охранную систему прямо на сайте, но в последствии мы поняли, что будет гораздо удобнее это делать в Telegram-боте, нежели на сайте, так как уведомления о

Ивл. № подл.	Подп. и дата	Взам. инв. №	Ивл. № дубл.	Подп. и дата

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



После того, как определились со структурой нашего сайта мы перешли к дизайну. Мы стремились сделать его простым, лаконичным и удобным для пользователя. Мы уделили особое внимание расположению каждого элемента, чтобы при входе на страницу пользователю сразу было понятно, где что находится. Именно благодаря простому дизайну человеку не придётся тратить много времени на поиск того или иного элемента. Большинство объектов размещены именно там, где их ожидают увидеть большинство пользователей, чтобы процесс взаимодействия был интуитивно понятен каждому. Также в футере нашего сайта мы разместили ссылку и QR-код на Telegram-бота для удобства пользователя.

					15.03.06.930000.000 ПЗ	Лист
Изм	Лист	№ докум.	Подп.	Дата		20

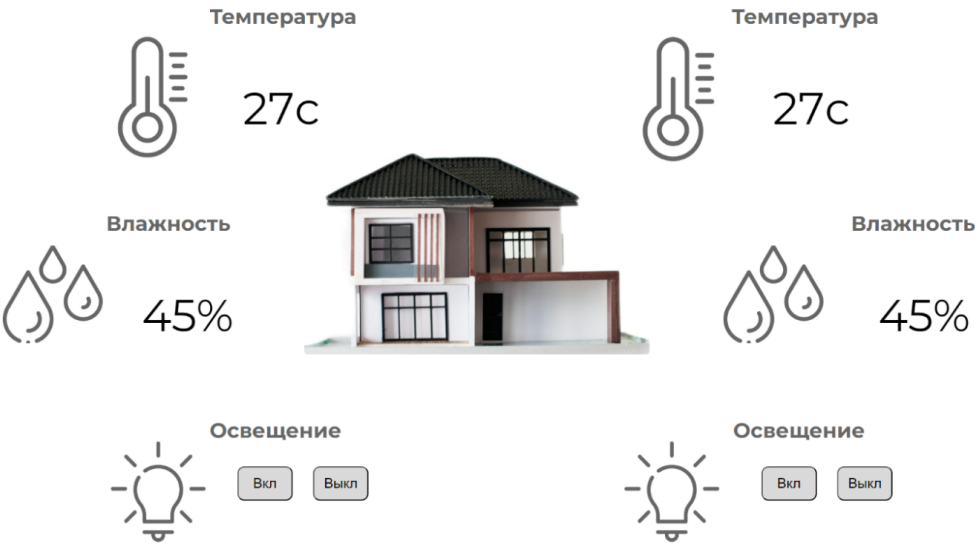
очень удобно работать командой, она позволяет нескольким пользователям работать над проектом одновременно, обмениваться комментариями и видеть изменения в реальном времени. Это делает процесс разработки более эффективным и удобным для командной работы. Также в Figma можно взять информацию об объекте (его размеры и позиционирование, цвет, шрифт, текстовые стили и эффекты), что очень удобно для прописывания стилей в CSS.

№
Е С Т О



1 этаж

2 этаж



Служба поддержки:

+7 989 526 49 67

ulyanakukovskaya@gmail.com

Управляй умным домом через Telegram-бот:

[Mesto_bot](#)

Рисунок 5.3 — Дизайн сайта.

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата

5.3 Разработка веб-страницы

Мы решили использовать для frontend-разработки редактор кода Sublime text, так как данное приложение является удобным и простым в применении. Для вёрстки сайта мы использовали HTML, а для прописывания стилей элементов - CSS. Мы выбрали именно данные языковые разметки, потому что они являются основополагающими технологиями веб-разработки, также изучение их принципов позволяет понять основы создания веб-страниц. Помимо этого HTML и CSS являются достаточно простыми для изучения и понимания, что делает их отличным выбором для начинающих frontend-разработчиков.

В HTML документе мы прописывали функции каждого из объектов, а также прописывали типы переменных. Наша страница содержит различные элементы, такие как header и footer, ссылки на стили, изображения, текстовые блоки, а также кнопки.

В заголовке страницы указана метаинформация, такая как кодировка символов, масштаб отображения и ссылки на внешние таблицы стилей:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="main.css">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100..900;1,100..900&display=swap" rel="stylesheet">
```

Рисунок 5.4 — тег head

Также в head указан заголовок веб-страницы, а также иконка, отображаемая во вкладке браузера:

```
<title>Место-умный дом мечты</title>
<link rel="icon" type="image/png" href="C:\Users\User\Desktop\Design\casito3.jpg">
</head>
```

Рисунок 5.5 — Заголовок веб-страницы

В теле нашей веб-страницы содержатся различные блоки, такие как header:

```
<header>

</header>
```

Рисунок 5.6 — header сайта

Также в body сайта находятся контейнеры с изображениями и текстом, информация о температуре, влажности и освещении, кнопки управления:

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата	15.03.06.930000.000 ПЗ					Лист
										22
Изм	Лист	№ докум.	Подп.	Дата						

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата
Изм	Лист	№ докум.	Подп.	Дата

```

<div class="container1">
  
  
  <div class='text'>Новый уровень комфорта и уюта</div>
</div>
<div class="container2">
  <div class='logo1'>1 этаж</div>
  <div class='logo2'>2 этаж</div>
  
  <div class="top">
    <div class='temperature'> Температура</div>
    <div class='humety'> Влажность</div>
    <div class='light'> Освещение</div>
    <div class='temperature2'> Температура</div>
    <div class='humety2'> Влажность</div>
    <div class='light2'> Освещение</div>
  </div>
  <div class="sensor-window">
    <h2></h2>
    <div id="sensor-data">27c</div>
  </div>
  <div class="sensor-window2">
    <div id="sensor-data2">27c</div>
  </div>
  <div class="sensor-window3">
    <div id="sensor-data3">45%</div>
  </div>
  <div class="sensor-window4">
    <div id="sensor-data4">45%</div>
  </div>
  
  
  
  
  
  
  <div class="top3">
    <div><button class='on1'>Вкл</button></div>
    <div><button class='on2'>Вкл</button></div>
    <div><button class='off1'>Выкл</button></div>
    <div><button class='off2'>Выкл</button></div>
  </div>
</div>

```

Рисунок 5.7 — основная часть сайта

В нашем коде мы использовали теги <div class="sensor-window», которые содержат информацию о датчиках и их показаниях.

В footer мы поместили контактную информацию, QR-код и ссылку на Telegram-бот для управления умным домом:

```

<footer>
  <div class='text2'>Служба поддержки:</div>
  <div class='text3'>+7 989 526 49 67</div>
  <div class='text4'>ulyanakukovskaya@gmail.com</div>
  
  <div class='text5'>Управляй умным домом через Telegram-бот:</div>
  <a href="https://t.me/Mestoo_bot"><u>Mestoo_bot</u></a>
</footer>

```

Рисунок 5.8 — footer

Также в коде присутствуют ссылки на изображения, которые доступны по указанным путям.

В документе CSS мы прописывали стили наших объектов. Мы брали информацию о каждом объекте из Figma, для ускорения процесса. Также для

header, body, footer мы прописывали параметры, которые применялись сразу ко всем объектам, находившимся в них:

```
body {
  font-family: Montserrat, sans-serif;
  margin: 0;
  padding: 0;
}
header {
  padding: 10px 0;
  position: relative;
}
```

Рисунок 5.9 — Стили body, header

```
footer {
  font-family: Montserrat, sans-serif;
  margin: 0;
  background-color: #333;
  top: 1950px;
  color: #fff;
  height: 200px;
  text-align: left;
  position: absolute;
  left: 0;
  width: 100%;
}
```

Рисунок 5.10 — Стили footer

В процессе разработки мы столкнулись с рядом проблем, так как мы с нуля учились создавать сайт, но несмотря на различные трудности, мы получили большое количество знаний и освоили новые языки программирования.

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата
Изм	Лист	№ докум.	Подп.	Дата

6 Телеграмм бот

Бот в Telegram — это программа, которая автоматизирует определенные задачи и взаимодействие с пользователями в мессенджере Telegram.5

Мы приняли решение воспользоваться телеграмм-ботом в связи с его быстрым и удобным функционалом для отправки и получения информации, как на этапе разработки, так и в процессе эксплуатации.

Плюсы бота которые мы тоже имели в виду при выборе решения. Это то что боты в телеграмме быстро работают и отвечают; ими удобно пользоваться, так как выбор текстовых команд ни у кого не вызывает затруднений; они не нуждаются в установке дополнительного ПО, поскольку любое взаимодействие с ботом осуществляется посредством мессенджера; не затрагивают личные данные без непосредственной команды пользователя и многое другое.

Наш телеграмм бот называется Место по ссылке: https://t.me/Mestoo_bot

Цель Место бота: позволяет тебе контактировать с различными функциями умного дома "Место".

6.1 Используемые материалы для создания телеграмм бота

Для понятия какие материалы использовать для создания самого бота с функциями которые мы хотели в него внедрить, они будут более подробно описаны далее:

Telegram Bot API (<https://core.telegram.org/bots/api>):

Это официальное API Telegram, предоставляющее возможность создания и взаимодействия с ботами в мессенджере Telegram. API предоставляет различные методы для отправки сообщений, управления подписчиками, работу с медиа-файлами и другие функции. 6

BotFather (<https://t.me/botfather>):

BotFather является официальным ботом Telegram, который помогает создавать и настраивать новых ботов. Он предоставляет удобный интерфейс для регистрации новых ботов, настройки их параметров, получения токенов доступа и управления командами. 6

Язык программирования Python (<https://www.python.org/>):

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата	Для понятия какие материалы использовать для создания самого бота с функциями которые мы хотели в него внедрить, они будут более подробно описаны далее:					
					Telegram Bot API (https://core.telegram.org/bots/api):					
					Это официальное API Telegram, предоставляющее возможность создания и взаимодействия с ботами в мессенджере Telegram. API предоставляет различные методы для отправки сообщений, управления подписчиками, работу с медиа-файлами и другие функции. 6					
					BotFather (https://t.me/botfather):					
					BotFather является официальным ботом Telegram, который помогает создавать и настраивать новых ботов. Он предоставляет удобный интерфейс для регистрации новых ботов, настройки их параметров, получения токенов доступа и управления командами. 6					
Инов. № подл.	Подп. и дата	Язык программирования Python (https://www.python.org/):								
Изм	Лист	№ докум.	Подп.	Дата	15.03.06.930000.000 ПЗ					Лист
										25

Python - это высокоуровневый язык программирования, который широко используется для разработки телеграм-ботов благодаря своей простоте, мощности и богатой экосистеме библиотек.7

Среда разработки PyCharm (<https://www.jetbrains.com/pycharm/>):

PyCharm - это интегрированная среда разработки (IDE) для языка программирования Python, предоставляющая широкие возможности для разработки, отладки и управления проектами. 8

Библиотека telebot для работы с Telegram API в Python (<https://github.com/eternnoir/pyTelegramBotAPI>):

Библиотека telebot - это набор инструментов для работы с Telegram Bot API на языке Python. Она облегчает взаимодействие с API, предоставляя удобные методы для отправки сообщений, обработки команд и работы с медиа-файлами.9

6.2 Функции телеграмм бота

Функции телеграмм ботов, в основном направлены на отправку сообщений, получение информации, обработка команд, уведомление и оповещения, интерактивные элементы как кнопки, работа с базой данных и интеграция с внешними сервисами.

Оценивая нашу целевую аудиторию и цель телеграмм бота, мы внедрили следующие функции для управления умным домом через мессенджер телеграмм:

/Menu (кнопка либо команда)

Если была вызвана команда, то бот будет выводить следующие:

Ручные функции

/temperaturenow - актуальные данные температуры дома

/humiditynow - актуальные данные уровня влаги во воздухе дома

/photocheck - Актуальные фото камер у входов дома

Автоматические функции

- Обнаружение посторонних лиц у входа - отправляется фото этих лиц

- Повышение уровня температуры дома

Если уже будет взаимодействие с кнопкой Menu, она предоставит остальные кнопки для действия

/start (кнопка либо команда)

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	сервисами.				
					Оценивая нашу целевую аудиторию и цель телеграмм бота, мы внедрили следующие функции для управления умным домом через мессенджер телеграмм:				
					/Menu (кнопка либо команда)				
					Если была вызвана команда, то бот будет выводить следующие:				
					Ручные функции				
					/temperaturenow - актуальные данные температуры дома				
					/humiditynow - актуальные данные уровня влаги во воздухе дома				
					/photocheck - Актуальные фото камер у входов дома				
					Автоматические функции				
					- Обнаружение посторонних лиц у входа - отправляется фото этих лиц				
					- Повышение уровня температуры дома				
					Если уже будет взаимодействие с кнопкой Menu, она предоставит остальные кнопки для действия				
					/start (кнопка либо команда)				

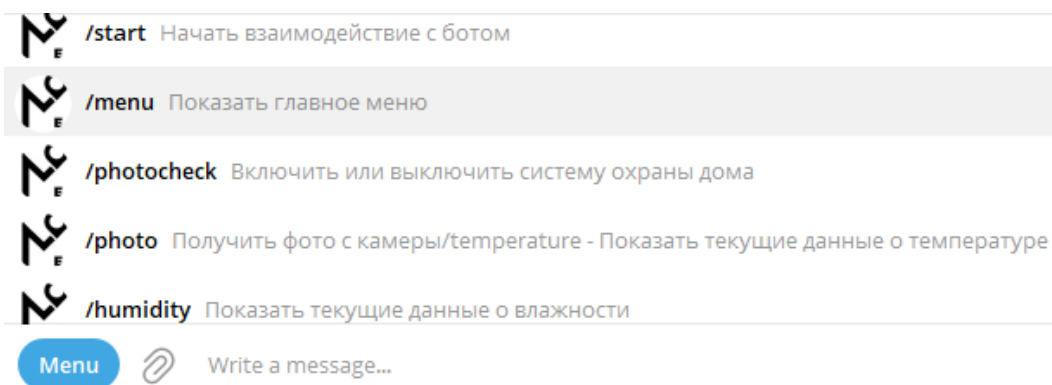


Рисунок 6.1 — Изображение меню через функцию кнопки

Для безопасности пользователя бот будет приветствовать его по имени и фамилии, используя данные, предоставленные в Telegram.

В базе данных бота хранятся идентификационные данные устройства умного дома и его владельца (ID пользователя Telegram, имя и фамилия зашифрованы). При вызове команды /start, если бот не распознает идентификационный номер запросившего, данные дома не будут отображаться при запросе.

Эта мера безопасности, чтобы предотвратить доступ к конфиденциальным данным дома без соответствующей идентификации пользователя.

/photo (кнопка либо команда)

Это для получения фотографии реального времени с камер находящие в устройстве умного дома.

Также фотографии отправляются автоматически совместно с уведомлением, при обнаружении посторонних лиц.

/photocheck (кнопка либо команда)

Эта функция для включения или выключения камер дома.

/humiditynow (кнопка либо команда)

Она отправляет актуальные данные датчиков устроенных в системе умного дома, данные именно уровня влажности воздуха.

Также, система при обнаружении уровня влажности менее 30% либо более 80%, отправляет уведомление о данной информации. Это и за того что слишком высокая влажность воздуха (более 60-70%) может способствовать развитию плесени, аллергическим реакциям и росту бактерий, что может негативно сказаться на здоровье. 10 А слишком низкая влажность (менее 30%) может вызывать сухость слизистых оболочек, раздражение глаз, кожи и дыхательных путей. 10

Инв. № подл.	Подл. и дата	Взам. инв. №	Инв. № дубл.	Подл. и дата	15.03.06.930000.000 ПЗ					Лист
										27
Изм.	Лист	№ докум.	Подп.	Дата						

/temperaturenow (кнопка либо команда)

Функция работает наподобие команды и кнопки /humiditynow. Различие заключается в том, что вместо данных уровня влажности воздуха, выводятся данные температуры в реальное время.

В целом, основная задача телеграм-бота заключается в обеспечении пользователей умного дома удобного способа взаимодействия с виртуальными функциями, позволяя им управлять своим домом через один мессенджер, таким как Telegram. Таким образом, нет необходимости загружать дополнительные приложения для мониторинга и управления.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	15.03.06.930000.000 ПЗ					Лист
										28
Изм.	Лист	№ докум.	Подп.	Дата						

7 Заключение

В ходе выполнения данной курсовой работы, были рассмотрены различные аспекты системы умного дома, начиная с анализа целевой аудитории проекта и заканчивая созданием бота в Telegram.

Анализ целевой аудитории позволил определить основные потребности и предпочтения пользователей, что является ключевым фактором при разработке и внедрении системы умного дома.

Создание системы «Умный дом» включало в себя разработку внешнего вида стенда, взаимодействие элементов системы, алгоритм работы и особенности датчиков. Всё это способствовало созданию функциональной и удобной для использования системы.

Использование системы компьютерного зрения в проекте открывает новые возможности для повышения уровня безопасности и управления различными устройствами в доме

Разработка web-сайта и создание бота в Telegram дополняют функционал системы умного дома, обеспечивая возможность управления и мониторинга через различные платформы.

В целом, данная работа позволила глубже изучить технологии систем умного дома, их применение и перспективы развития. Реализация данных технологий способствует созданию современного и инновационного жилищного пространства, соответствующего потребностям современного общества.

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата						Лист
										29
Изм	Лист	№ докум.	Подп.	Дата	15.03.06.930000.000 ПЗ					

8 Перечень использованных информационных ресурсов

- а) Российский рынок умного дома в 2022 году. Продажи, игроки, перспективы – URL: <https://mobile-review.com/all/articles/analytics/rossijskij-rynok-umnogo-doma-v-2022-godu-prodazhi-igroki-perspektivy>
- б) Интернет-Маркетинг для проекта «Умный дом», основываясь на исследованиях яндекса – URL: <https://akiwa.ru/blog/internet-marketing-dlya-proekta-umnyy-dom-osnovyvayas-na-issledovaniyakh-ya-dek-sa>
- в) Умный дом Алиса – URL: <https://yandex.ru/alice/smart-home>
- г) Korolab умный дом – URL: <https://korolab.ru>
- д) Что такое Telegram и как им пользоваться. GB.RU. – URL: <https://gb.ru/blog/cto-takoe-telegram/#::text=>
- е) Всё, о чём должен знать разработчик Телеграм-ботов. (2021). Взято с Хабр. – URL: <https://habr.com/ru/articles/543676/>

ж) Amazon Web Services (AWS): "What is Python?" Взято с Amazon Web Services. URL: <https://aws.amazon.com/ru/what-is/python/>

и) "Что такое PyCharm?" Взято с Skyeng. – URL: <https://skyeng.ru/magazine/cto-takoe-pycharm/>

к) "Какую библиотеку на Python выбрать для создания телеграм-бота?" Взято с Хабр. – URL: <https://habr.com/ru/companies/otus/articles/771110/>

л) "Как влияет влажность воздуха на организм человека": Par-Tuman.ru – URL: <https://par-tuman.ru/sferi-primeneniya/180-kak-vliyaet-vlazhnost-vozdukha-na-organizm-cheloveka>

Инд. № подл.	Подп. и дата	Взам. инв. №	Инд. № дубл.	Подп. и дата					
Изм	Лист	№ докум.	Подп.	Дата	15.03.06.930000.000 ПЗ				
					Лист 30				