

Всероссийский конкурс научно-технологических проектов “Большие вызовы”

Направление: Умный город и безопасность

**РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ КОНТРОЛЯ  
МИКРОКЛИМАТА ПРОМЫШЛЕННЫХ ТЕПЛИЦ**

**Работу выполнил:**

Зезюля Кирилл Романович

**Руководитель:**

Крестниковская Екатерина Павловна

Педагог дополнительного образования

ГАУДО "ЦЦО "IT-Куб"

Брянск 2023

## СОДЕРЖАНИЕ

|  |  |
|--|--|
| <b>ВВЕДЕНИЕ</b> .....                                      | <b>3</b>                               |
| <b>ТЕОРЕТИЧЕСКАЯ СЕКЦИЯ</b> .....                          | <b>4</b>                               |
| <b>ГЛАВА 1 ОБЗОРНО-АНАЛИТИЧЕСКАЯ ЧАСТЬ</b> .....           | <b>4</b>                               |
| 1.1 Анализ целевой аудитории.....                          | 4                                      |
| 1.2 Опрос целевой аудитории.....                           | 5                                      |
| <b>ГЛАВА 2. Маркетинговое исследование</b> .....           | <b>7</b>                               |
| 2.1 Методика маркетингового исследования (админ) .....     | <b>Ошибка! Закладка не определена.</b> |
| 2.2 Методика маркетингового исследования (клиент) .....    | <b>Ошибка! Закладка не определена.</b> |
| <b>ГЛАВА 3. РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ</b> .....     | <b>8</b>                               |
| 3.1 Функциональные характеристики .....                    | 8                                      |
| 3.2 Исследование рынка аналогов и альтернатив .....        | 8                                      |
| 3.3 Описание технологии.....                               | <b>Ошибка! Закладка не определена.</b> |
| 3.4 Разработка программной части .....                     | 10                                     |
| <b>ЗАКЛЮЧЕНИЕ</b> .....                                    | <b>16</b>                              |
| <b>СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ</b> .....                | <b>17</b>                              |
| <b>ПРИЛОЖЕНИЕ 1 Общий функционал работы продукта</b> ..... | <b>18</b>                              |

## ВВЕДЕНИЕ

Любое тепличное хозяйство имеет в штате овощеводов, которые ухаживают за растениями. Каждый день они подвязывают, поливают, подкармливают тысячи кустов, и это всё в комбине зонах при достаточно высокой температуре.

Моё мобильное приложение для контроля микроклимата промышленных теплиц получает данные в реальном времени от датчиков и камер в теплице, может включить определённые помпы для полива растений, отслеживает условия климата, установленные параметры для идеальных условий, получает уведомления и подаёт сигнал, если любые изменения происходят в окружающей среде.

**Актуальность проекта:** работа на данную тему является актуальной, так как большое количество времени овощевод занят подкормкой растений, сбором показателей с датчиков вместо того, чтобы заняться более важными делами.

**Проблема:** собирать измерения теплиц вручную сложно и трудозатратно, по этой причине сотрудники используют гаджеты. Гаджетов много, показатели разные, они находятся на разных накопителях, поэтому приходится оформлять отчеты в едином документе на компьютере.

**Цель:** разработать мобильное приложение для работы в промышленных теплицах, предназначенное для сбора показателей со всех датчиков на экране одного устройства, составление отчета в виде таблицы с пояснениями на основе полученных данных.

**Задачи:**

- Описать целевую аудиторию моего проекта.
- Выявить и исследовать аналоги и возможные альтернативы.
- Выполнить поставленную цель и сделать выводы.

**Заказчик:** тепличные хозяйства, агрохолдинги и садовые фермы.

## ТЕОРЕТИЧЕСКАЯ СЕКЦИЯ

### ГЛАВА 1 ОБЗОРНО-АНАЛИТИЧЕСКАЯ ЧАСТЬ

#### 1.1 Анализ целевой аудитории

Целевая аудитория – агрохолдинги

Основной сегмент ЦА — это **тепличные хозяйства**. При создании мобильного приложения ориентировались на **огромные промышленные холдинги**. Общий процент 67%. Им нужна качественная автоматизированная система с минимальными затратами.



Рис 1. Анализ целевой аудитории

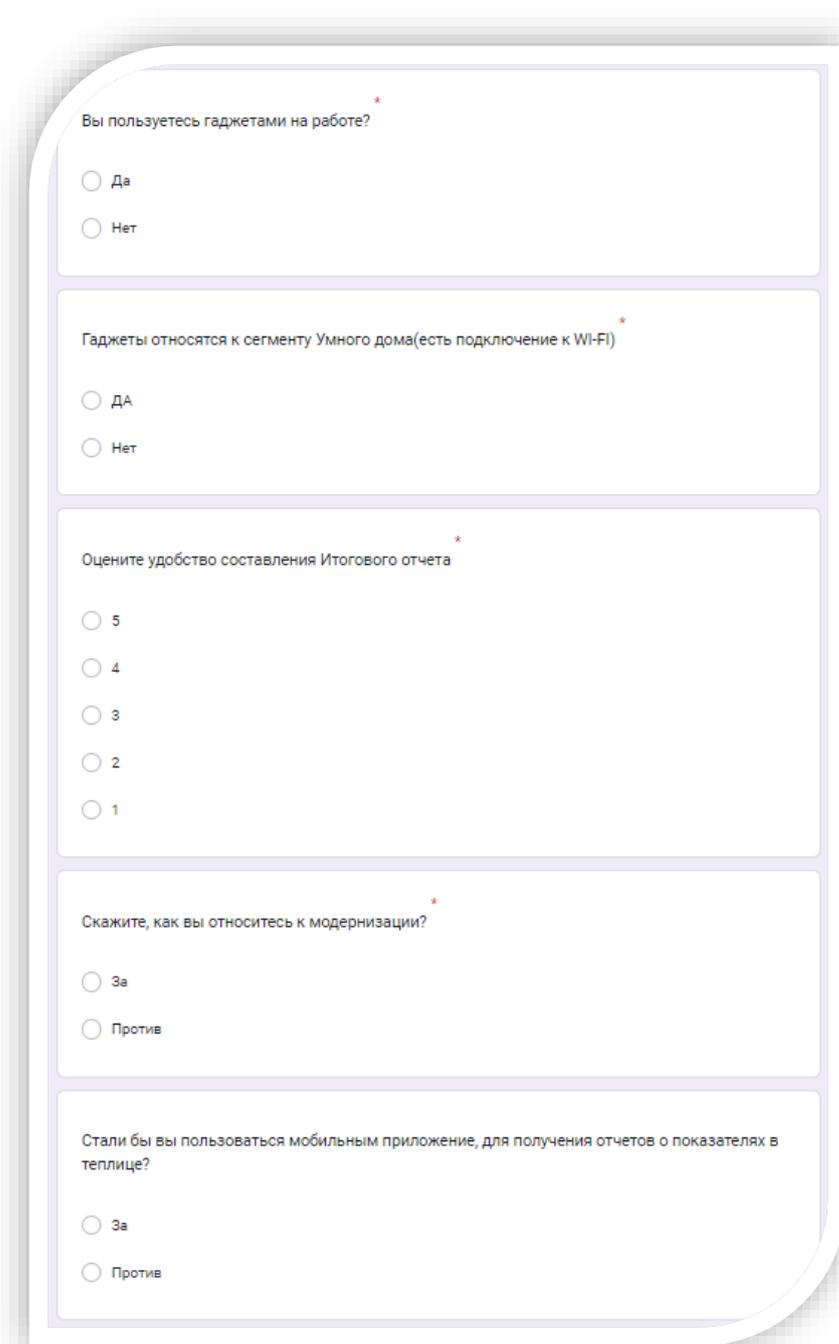
**Заказчик** – агрохолдинги. Собирать измерения теплиц вручную сложно и трудозатратно, по этой причине сотрудники используют гаджеты. Гаджетов много, показатели разные, поэтому приходится оформлять отчеты в едином документе на компьютере.

**Задача:** создать мобильное приложение для работы с промышленными теплицами.

Я предложил разработать мобильное приложение для телефонов, которое будет собирать показатели со всех гаджетов на экране одного устройства, а позже выдавать отчет в виде таблицы с пояснениями.

## 1.2 Опрос целевой аудитории

Был составлен список основных преимуществ “СледиЗаТеплицей” на бесплатной основе. Было добавлено поле о том, сколько опрошиваемые готовы платить за подписку на приложение.



Вы пользуетесь гаджетами на работе? \*

Да

Нет

Гаджеты относятся к сегменту Умного дома(есть подключение к WI-FI) \*

Да

Нет

Оцените удобство составления Итогового отчета \*

5

4

3

2

1

Скажите, как вы относитесь к модернизации? \*

За

Против

Стали бы вы пользоваться мобильным приложением, для получения отчетов о показателях в теплице?

За

Против

## Рис 2. Форма опроса целевой аудитории

Был составлен анонимный опрос на Google формах и распространен через форумы и мессенджеры. Опрос длился 1 месяц, в итоге приняли участие 27 человек.

При составлении опроса учитывались моменты за/против.

Вы пользуетесь гаджетами на работе?

- Да(100%)
- Нет (0%)

Гаджеты относятся к сегменту Умного дома(есть подключение к WI-FI) ?

- Да(83%)
- Нет (17%)

Оцените удобство составления Итогового отчета?

- 5 (0%)
- 4(0%)
- 3(66%)
- 2(17%)
- 1(17%)

Скажите, как вы относитесь к модернизации?

- За(100%)
- Против(0%)

Стали бы вы пользоваться мобильным приложением, для получения отчетов о показателях в теплице?

- За(100%)
- Против(0%)

Исходя из запросов аудитории, было принято решение – разработать мобильное приложение.

## ГЛАВА 2. Маркетинговое исследование

### 2.1 Методика маркетингового исследования (клиент)

Участники: 27 представителей ЦА.

Опрос и демонстрация проводилась через мессенджеры и форумы.

Демонстрация пути пользователя и прототипа мобильного приложения.

Вопросы:

1. Вас заинтересовал функционал мобильного приложения?
2. Стали бы вы пользоваться приложением в рабочее время?
3. Как вы оцениваете полезность мобильного приложения?

Результат

Участников заинтересовал прототип мобильного приложения. 17 из 27 респондентов готовы скачать моё мобильное приложение.

Запросы:

- Так как работает с организацией, желательно чтобы пользователей регистрировал админ и сразу загружались данные.
- То есть нужно разделить вход на админа и агронома.
- В целом приложение заинтересовало пользователей, есть запрос на разработку.

Анализ конкурентов нужно проводить регулярно, не только на старте запуска компании. Результаты проведенного исследования подскажут, в каком направлении развиваться дальше, чтобы клиенты выбирали только вас. Недаром же говорят: посмотри на конкурентов и сделай лучше – отличная формула для любого бизнеса.

## ПРАКТИЧЕСКАЯ СЕКЦИЯ

### ГЛАВА 3. РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ

#### 3.1 Функциональные характеристики

Мобильное приложение должно отвечать следующим характеристикам:

- Подходит для большинства мобильных устройств
- Можно работать на планшете
- Просто установить
- Выбрана бесплатная среда разработки Android, язык

программирования JAVA

Требования к функциональным характеристикам:

1. Авторизации администратора по почте и паролю, регистрация работников

2. Работа офлайн/онлайн с БД

3. Широкий функционал работы с БД

4. Поиск по заказам

5. Специализация: кофейня

6. Статистика по заказам: ТОП продаж

7. Работа с СБП, учет переводов онлайн на карту

8. Функционал Онлайн/Офлайн

9. Окна: регистрация/автоматизация, работа с заказами, карточка пользователя, статистика по заказам

10. Сканер QR-кода

#### 3.2 Исследование рынка аналогов и альтернатив

В результате анализа были обнаружены три конкурирующие программы(FARMHAND, APPLY YOURSELF APP, VISION CLOUD), которые сейчас популярны у представителей агрохолдингов. У каждого ПО были выявлены общие преимущества: Работа с датчиками, облачное хранение данных,

интуитивно понятный интерфейс, работа без интернета . Каждая из них имеет недостатки, которые я взял в работу для своего приложения.



Рис 3. Альтернативы

### FARMHAND

Минусы:

1. Недостаточное количество функций работы с датчиками.
2. Малое количество скачиваний на Google Play

### APPLY YOURSELF APP

Минусы:

1. Устаревший интерфейс программы.
2. Недостаточное количество функций работы с датчиками.

### VISION CLOUD

Минусы:

1. Небольшое число возможностей для управления через интернет.
2. Нет функции работы оффлайн.

### 3.3 Описание технологии

При реализации проекта за основу был взят функционал ранее существующих продуктов. В разработки были использованы следующие технологии:

- Material Design — дизайн-система для создания интерфейсов программного обеспечения и приложений, разработана компанией Google
- Мобильное приложение разработано в среде Android Studio, язык разметки XML, языки программирования Kotlin и Java
- Регистрация и облачное хранение данных в Firebase

- Офлайн базы данных SQLite

Прототип мобильного приложения

1-вход/регистрация

2-кабинет пользователя

3-добавление/поиск теплицы.

### Общая схема работы мобильного приложения

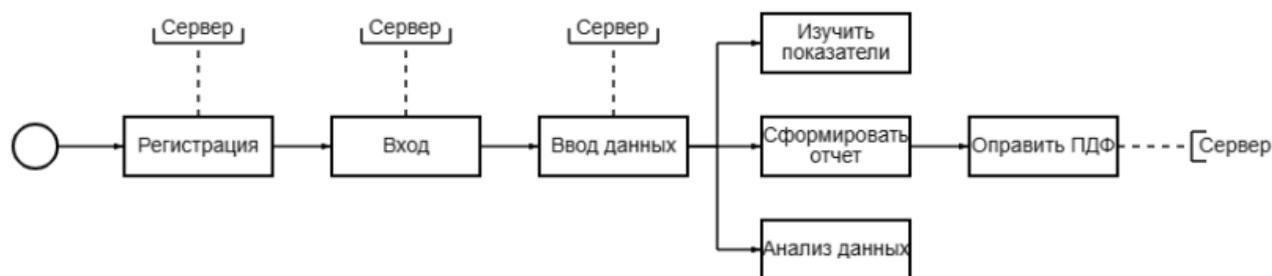


Рис. 4 Путь Пользователя

Прототип мобильного приложения (Приложение 1)

1- регистрация пользователя по почте и паролю.

2- создание виртуальной теплицы

3- изменение состояний различных нагрузок, отображение показателей датчиков

### 3.4 Разработка программной части

Мобильное приложение создавалось на базе функционала аналогов, было принято решение добавить ряд особенностей, которые способны выделить продукт на рынке мобильных приложений. С учетом требований к функциональным характеристикам было разработано техническое задание.

Перед мной поставлено несколько задач: реализовать функционал связи мобильного приложения с wi-fi модулем, посредством протокола http (рисунок 5), разработать интерфейс и функциональную часть для изменения состояния полезной нагрузки и отображением показателей датчиков.

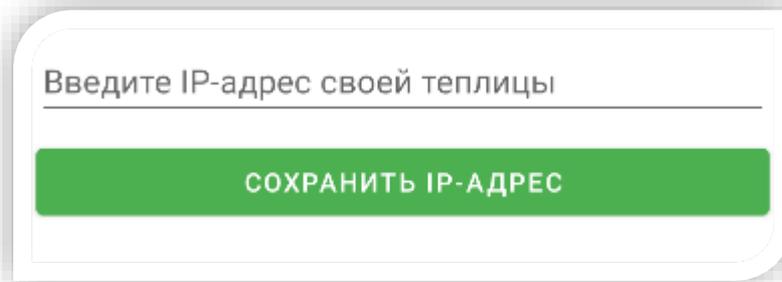


Рис. 5. Окно подключения wi-fi модуля

Создано окно для отображения показаний датчиков (например датчик температуры) (рисунок 6).



Рис. 6. Показатели датчика

Также создано окно для изменения состояния полезной нагрузки (например помпы, светодиоды, реле и т.д.) (рисунок 7).

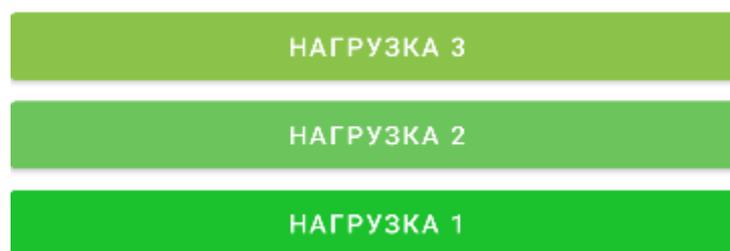


Рис. 6. Включение и выключение нагрузки

Макет главного окна разработан с учетом запросов пользователей относительно функциональности интерфейса. Переход между окнами реализован путем скользящего движения пальца с левой стороны экрана (рисунок 8). Используются векторные картинки в формате xml.

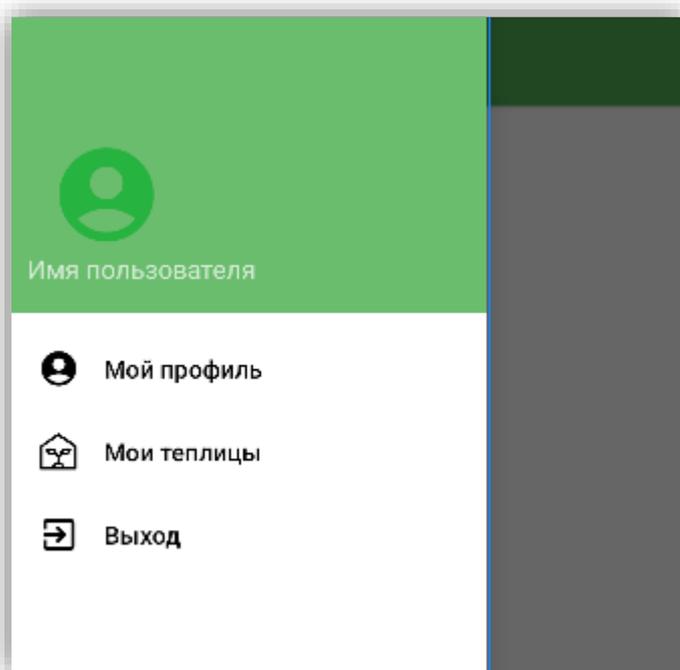


Рис 8. Функциональный интерфейс

Интерфейс предполагает наличие еще ряда неупомянутых функций, стандартных для любого мобильного приложения.

Таблица 1. Анализ конкурентов

| <b>Название</b>                       | <b>Преимущества</b>   | <b>Недостатки</b>   |
|---------------------------------------|---|---|
| Проект<br>(СледиЗаТеплицей)           | управления фермой из<br>любого места                                | Необходима<br>установка                                   |
| Сбор в базу данных                    | Возможность введения<br>ограничений для<br>обеспечения безопасности | Дополнительные<br>затраты на<br>аппаратное<br>обеспечение |
| Вести записи на<br>бумаге             | Простота в обслуживании и<br>низкая квалификация<br>рабочих         | Большая<br>вероятность<br>потерять данные                 |
| Вести записи в<br>электронной таблице | Интуитивно понятный<br>концепт                                      | Слабо<br>структурированная<br>модель данных               |
| Анализ с помощью<br>нейросети         | Очень удобно и автономно.   | Не разработано  |

Какие выводы можно сделать из полученной информации? Составить характеристику по каждому конкуренту. Конкуренты с высокой долей рынка и высокими темпами роста задают правила игры на рынке. Конкуренты с отрицательными или невысокими темпами роста могут служить хорошим источником для роста бизнеса. Конкуренты с высокими темпами роста могут рассказать об успешных техниках продаж.

## ГЛАВА 4. СОЗДАНИЕ СХЕМЫ ДЛЯ СВЯЗИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ С WI-FI МОДУЛЕМ

Для того, чтобы продемонстрировать работу своего приложения, я создал пример системы, находящейся в тепличном хозяйстве.

Для создания схемы устройства я воспользовался программой “Fritzing” (рисунок 9).

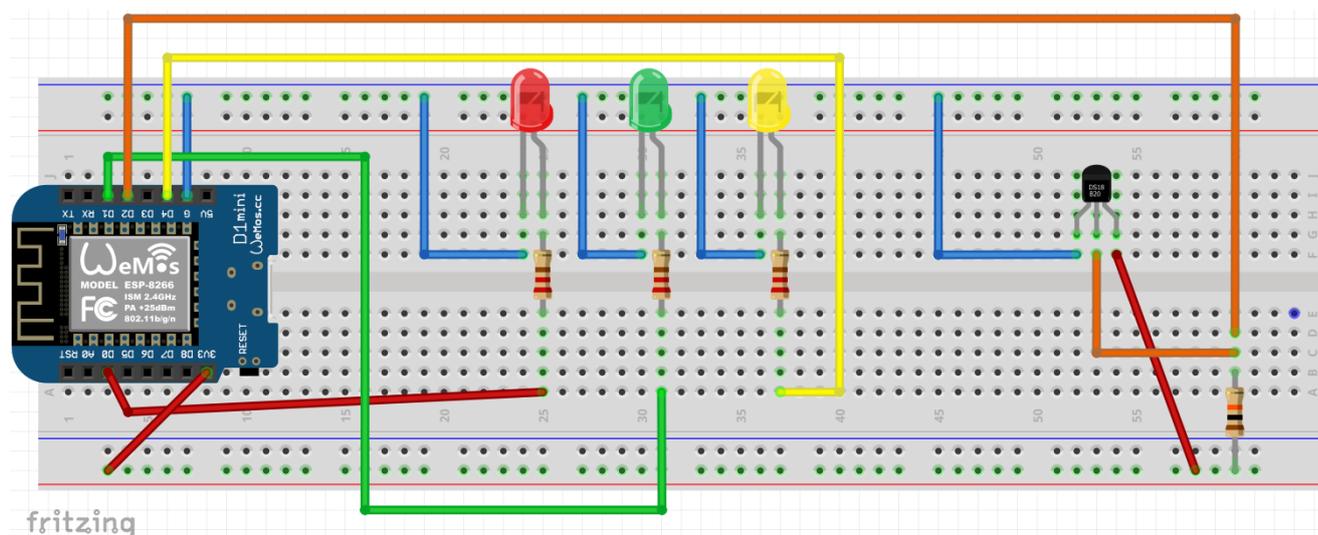


Рис 9. Схема

Чтобы создать программу для связи wi-fi модуля с мобильным приложением, я воспользовался средой разработки Arduino IDE.

В ней я написал код для связи с приложением с помощью http протокола(листинг 2).

Потом загрузил код в wi-fi модуль и ввёл его IP-адрес поле ввода.

Далее я рассчитал все расходы на эту установку(таблица 2).

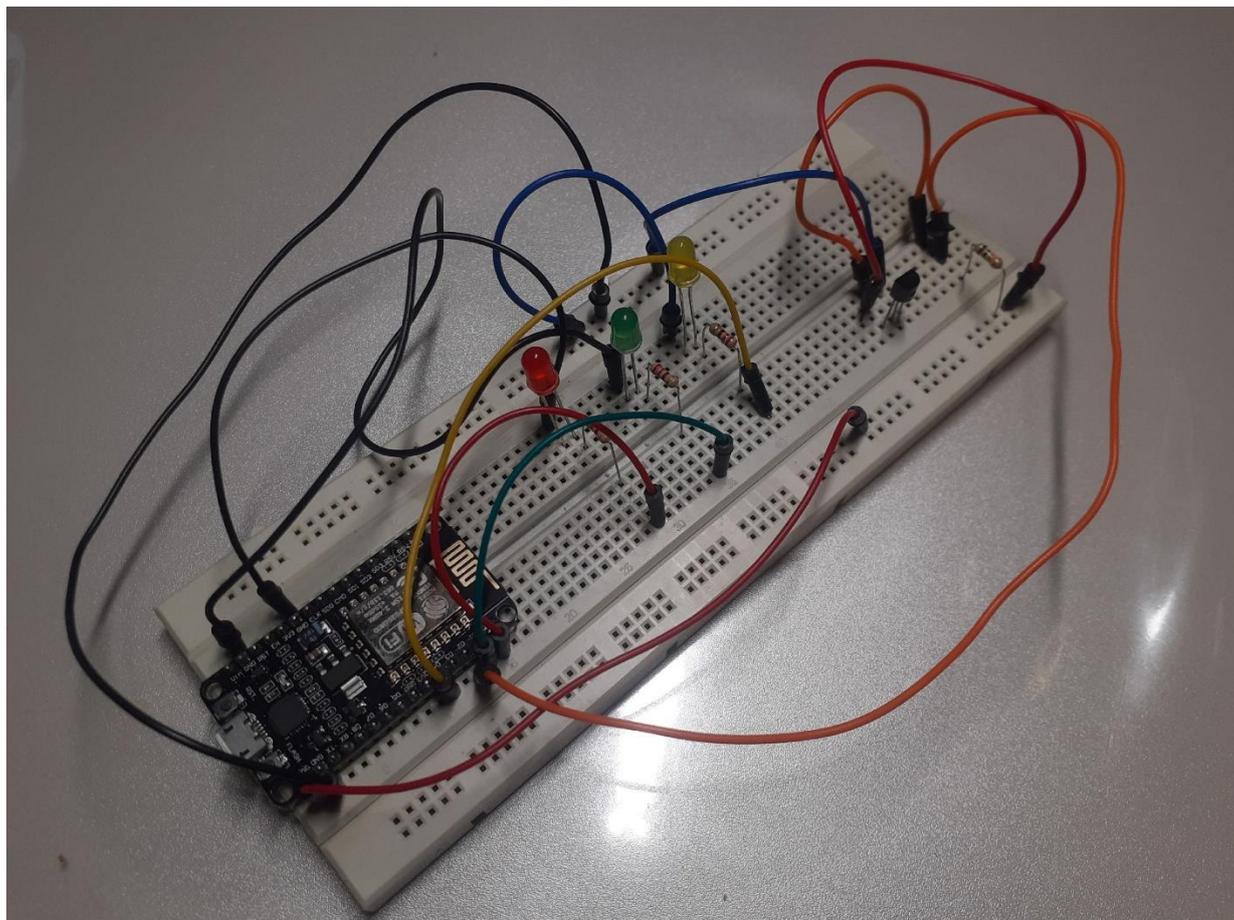
Таблица 2. Расходы на систему.

| Название                   | Количество | Цена | Общая стоимость |
|----------------------------|------------|------|-----------------|
| Доска для прототипирования | 1          | 90   | 90              |
| wi-fi модуль ESP8266       | 1          | 350  | 350             |
| Резисторы 100 Ом           | 3          | 0,5  | 1,5             |
| Резистор 10 КОм            | 1          | 0,5  | 0,5             |

|            |    |     |     |
|------------|----|-----|-----|
| Светодиоды | 3  | 1,5 | 4,5 |
| Провода    | 12 | 0,2 | 2,4 |

Суммарно я затратил – 448,9 рублей.

### ФОТО СХЕМЫ



## ЗАКЛЮЧЕНИЕ

Работа над приложением подтвердила важность целого ряда этапов в проектировании и создании приложений: структурной проработки и пользовательских сценариев, «бумажного» дизайна, анализа ошибок и удачных находок конкурентов.

В ходе работы над проектом было создано android-приложение на языке java в среде разработки Android Studio. Были изучены особенности различных интегрированных сред разработки мобильных приложений и разработка в Android Studio.

Поставленные цели и задачи были выполнены успешно. В планах дальнейшее развитие, модернизация и оптимизация приложения.

На основе полученных знаний было разработано многоэкранное приложение на языке Java. Приложение корректно отображается на платформе Android.

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Нужный, А. М. Разработка мобильных приложений на языке Java с использованием Android Studio: учебное пособие / А.М. Нужный, Н.И. Гребенникова, В. В. Сафронов. — Воронеж: ВГТУ, ЭБС АСВ, 2020. — 93 с. — ISBN 978-5-7731-0906-8.
2. Монахов, В. В. Язык программирования Java и среда NetBeans : учебное пособие / В. В. Монахов. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 450 с. — ISBN 978-5-4497-0923-3.
3. Вязовик, Н. А. Программирование на Java : учебное пособие / Н. А. Вязовик. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 601 с. — ISBN 978-5-4497-0852-6.
4. Свистунов, А. Н. Построение распределенных систем на Java : учебное пособие / А. Н. Свистунов. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 316 с. — ISBN 978-5-4497-0940-0.
5. Семакова, А. Введение в разработку приложений для смартфонов на ОС Android : учебное пособие / А. Семакова. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 102 с. — ISBN 978-5-4497-0892-2.
6. Семакова, А. Введение в разработку приложений для смартфонов на ОС Android : учебное пособие для СПО / А. Семакова. — Саратов : Профобразование, 2021. — 102 с. — ISBN 978-5-4488-0994-1.
7. Стрельцов, В. А. Полезный смартфон и планшет на Android / В. А. Стрельцов, М. А. Финкова, Р. Г. Прокди. — Санкт-Петербург : Наука и Техника, 2016. — 304 с.
8. ESP8266 WiFi Android Controller [Электронный ресурс]. – URL: <https://neco-desarrollo.es/2021/06/esp8266-wifi-android-controller> (дата обращения: 20.02.2023).

## **ПРИЛОЖЕНИЕ 1**

Общий функционал работы продукта



Имя пользователя



Мой профиль



Мои теплицы



Выход

Введите IP-адрес своей теплицы

---

СОХРАНИТЬ IP-АДРЕС

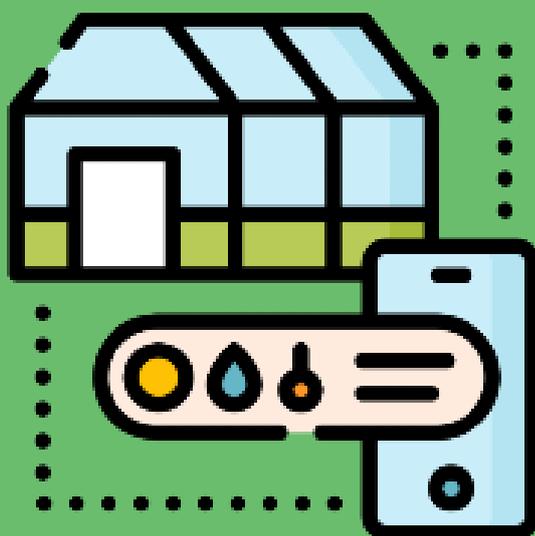
## Температура в теплице

0С°

НАГРУЗКА 3

НАГРУЗКА 2

НАГРУЗКА 1



СледиЗаТеплицей

## Листинг 1. Код мобильного приложения.

```
class GreenhouseActivity: AppCompatActivity() {
    private lateinit var request: Request
    private lateinit var binding: ActivityMainBinding
    private lateinit var pref: SharedPreferences
    private val client = OkHttpClient()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        pref = getSharedPreferences("MyPref", MODE_PRIVATE)
        onClickSaveIp()
        getIp()
        binding.apply {
            bLed1.setOnClickListener(onClickListener())
            bLed2.setOnClickListener(onClickListener())
            bLed3.setOnClickListener(onClickListener())
        }
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.main_menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        if(item.itemId == R.id.sync) post("temperature")
        return true
    }

    private fun onClickListener(): View.OnClickListener{
        return View.OnClickListener {
            when(it.id) {
                R.id.bLed1 -> { post("led1") }
                R.id.bLed2 -> { post("led2") }
                R.id.bLed3 -> { post("led3") }
            }
        }
    }

    private fun getIp() = with(binding){
        val ip = pref.getString("ip", "")
        if(ip != null){
            if(ip.isNotEmpty()) edIp.setText(ip)
        }
    }

    private fun onClickSaveIp() = with(binding){
        bSave.setOnClickListener {
            if(edIp.text.isNotEmpty()) saveIp(edIp.text.toString())
        }
    }

    private fun saveIp(ip: String){
        val editor = pref.edit()
        editor.putString("ip", ip)
        editor.apply()
    }
}
```

```

private fun post(post: String){
    Thread{

        request =
Request.Builder().url("http://${binding.edIp.text}/${post}").build()
        try {
            var response = client.newCall(request).execute()
            if(response.isSuccessful){
                val resultText = response.body()?.string()
                runOnUiThread {
                    val temp = resultText + "C°"
                    binding.tvTemp.text = temp
                }
            }
        } catch (i: IOException){

        }

    }.start()
}
}

```

Листинг 2. Код для wi-fi модуля.

```

#include <ESP8266WiFi.h>
#include <OneWire.h>
#include <DallasTemperature.h>

bool led1IsOn = false;
bool led2IsOn = false;
bool led3IsOn = false;
const int LED_0 = 16;
const int LED_1 = 5;
const int LED_2 = 2;

const int oneWireBus = 4;
const char* ssid = "Your SSID";
const char* password = "Your password";
float temperatureC = 0.0f;
// Create an instance of the server
// specify the port to listen on as an argument
WiFiServer server(80);

```

```
OneWire oneWire(oneWireBus);
DallasTemperature sensors(&oneWire);

void setup() {

  Serial.begin(115200);
  pinMode(LED_0, OUTPUT);
  pinMode(LED_1, OUTPUT);
  pinMode(LED_2, OUTPUT);
  delay(1000);
  digitalWrite(LED_0, HIGH);
  digitalWrite(LED_1, HIGH);
  digitalWrite(LED_2, HIGH);
  delay(1000);
  digitalWrite(LED_0, LOW);
  digitalWrite(LED_1, LOW);
  digitalWrite(LED_2, LOW);
  delay(10);
  sensors.begin();
  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
```

```

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

// Start the server
server.begin();
Serial.println("Server started");

// Print the IP address
Serial.println(WiFi.localIP());
}

void loop() {
  // Check if a client has connected
  sensors.requestTemperatures();
  WiFiClient client = server.available();
  if (!client) {
    return;
  }
  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }
}

```

```

// Read the first line of the request
String req = client.readStringUntil('\r');
Serial.println(req);
// Match the request
controller(req, client);
client.flush();
// Prepare the response
String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n";
s += temperatureC;
// Send the response to the client
client.print(s);
delay(1);
Serial.println("Client disconnected");

// The client will actually be disconnected
// when the function returns and 'client' object is destroyed
}
void controller(String req, WiFiClient client){
  if (req.indexOf("/temperature") != -1){
    temperatureC = sensors.getTempCByIndex(0);
    Serial.println("Showing temperature");
  } else if(req.indexOf("/led1") != -1){
    setLedState(LED_0, led1IsOn);
    led1IsOn = !led1IsOn;
  } else if(req.indexOf("/led2") != -1){
    setLedState(LED_1, led2IsOn);
    led2IsOn = !led2IsOn;
  } else if(req.indexOf("/led3") != -1){

```

```
    setLedState(LED_2, led3IsOn);
    led3IsOn = !led3IsOn;
} else {
Serial.println("invalid request");
client.stop();
return;
}
}

void setLedState(int led, bool state){
if(!state){
    digitalWrite(led, HIGH);
} else {
    digitalWrite(led, LOW);}
}
```