ЧАСТНА ПРОФЕСИОНАЛНА ГИМНАЗИЯ ЗА ДИГИТАЛНИ НАУКИ „СОФТУНИ БУДИТЕЛ“ , гр. София

**ДИПЛОМЕН ПРОЕКТ**

На Иван Иванов Илиев

ученик/ученичка от XII клас

професия-код: **481030, „Приложен програмист“**

специалност-код: **4810301, „Приложно програмиране“**

**Тема: Изработка на комбинация от IoT устройство и мобилно приложение за проследяване състоянието на дадено растение „Botanica Wellness”**

**Ръководител-консултант: Мартин Янев**

Сесия: май-юни 2024г.

Дата: 5.04.2024г.

**СЪДЪРЖАНИЕ**

[1. Увод 4](#_Toc165259291)

[1.1. Описание 4](#_Toc165259292)

[1.2. Мотивация 5](#_Toc165259293)

[1.3. Основна цел 6](#_Toc165259294)

[1.4. Очаквания 7](#_Toc165259295)

[1.5. Предизвикателства 8](#_Toc165259296)

[2. Изложение 9](#_Toc165259297)

[2.1. Проучвателна част 9](#_Toc165259298)

[2.2. Изисквания към програмния продукт 10](#_Toc165259299)

[2.2.1. Поставени задачи 10](#_Toc165259300)

[2.3. Същинска част на проекта 12](#_Toc165259301)

[2.3.1. IoT устройства 12](#_Toc165259302)

[2.3.2. Мобилен Клиент 13](#_Toc165259303)

[2.3.3. BWS (Botanica Wellness Server) 15](#_Toc165259304)

[2.4. Описание на използвани софтуерни технологии 16](#_Toc165259305)

[2.4.1. Уеб приложение 16](#_Toc165259306)

[2.4.2. Мобилно native приложение 17](#_Toc165259307)

[2.4.3. Kotlin 19](#_Toc165259308)

[2.4.4. WebView 20](#_Toc165259309)

[2.4.5. HTML 22](#_Toc165259310)

[2.4.6. CSS 23](#_Toc165259311)

[2.4.7. JavaScript 24](#_Toc165259312)

[2.4.8. Apexcharts 25](#_Toc165259313)

[2.4.9. PHP 26](#_Toc165259314)

[2.4.10. Custom API 27](#_Toc165259315)

[2.4.11. PostgreSQL 28](#_Toc165259316)

[2.4.12. NGINX 29](#_Toc165259317)

[2.4.13. Софтуер за IoT устройства 31](#_Toc165259318)

[2.5. Описание на използвани хардуерни технологии 33](#_Toc165259319)

[2.5.1. ESP32 Chipset 33](#_Toc165259320)

[2.5.2. DHT11 34](#_Toc165259321)

[2.5.3. BH1750 sensor 35](#_Toc165259322)

[2.5.4. Dell PowerEdge R720 35](#_Toc165259323)

[2.6. Ръководство на потребителя 36](#_Toc165259324)

[3. Заключение 45](#_Toc165259325)

[4. Информационни източници 46](#_Toc165259326)

[5. Приложения 47](#_Toc165259327)

# 

# **Увод**

## **Описание**

Живеем във времена на бързо сменящи се тенденции, бързо развиващи се технологии и бърз начин на живот. Напоследък нашумяла тема и водеща тенденция в земеделието и хоби градинарството е Precision Agriculture (PA). PA се цели да оптимизира ефективността на отглеждане на земеделски посеви и/или домашни растения, производителността на земеделските посеви и житни култури, да намали екологичното въздействие и рисковете както от доходно, така и от екологично гледна точка. Едно от основните приложения на PA, което се основава на екологичен аудит, е контролът на епидемичните заболявания. Епидемичните заболявания имат сериозни последици върху производството на култури. Ключовият фактор при епидемичните заболявания e измененията в климата, която настъпват неочаквано във времето и пространството, което прави последствията им по-сериозни.

Internet of Things (IoT) се разглежда като последната дума във внедряването на разпределени системи за мониторинг и контрол в различни области на приложение. Системата за мониторинг на Botanica Wellness на база IoT се стреми да използва технологията на безжични сензорни мрежи – Wireless Sensor Network (WSN) и е достъпна чрез интернет за приложения в PA, като контрол на епидемичните заболявания. Системата за управление на растителните заболявания на база IoT има за цел да постигне устойчиво развитие на земеделието. Тази система е достатъчно обща, за да се използва с множество растителни заболявания, където софтуерната архитектура на приложението може да управлява различни модели на растителни заболявания. Въпреки че платформата на приложението се базира на стандартен безжичен комуникационен слой, необходим е внимателен дизайн на системата, тъй като изискванията към платформата са много строги.

В основата си приложението Botanica Wellness е система за мониторинг и контрол на заболявания по растенията за домашна и търговска употреба, базирана на IoT с отворен код. Приложението се състои от три основни компонента. IoT устройства, BWS (Botanica Wellness Server) и мобилен андроид клиент. Самият мобилен клиент има различни функционалности за потребителите, които могат да задоволят всяко тяхно желание(проследяване на състоянието на растението в реално време, проследяване на състоянието на растението за даден период от време, задаване и запазване на пресети и референтни стойности за различни видове растения и други). Понеже платформата разчита на собствен бекенд и не се използват външни дистрибутори на услуги имаме сравнително сигурна и защитена платформа от страна на данните на потребителите.

## **Мотивация**

Проектът Botanica Wellness е една демонстрация на развитието и бъдещето на земеделските технологии, които се пренасят и в домашна среда. Основната мотивация за написването му е реалната липса на подобно решение, което да е едновременно с отворен код, да е висококачествено с ниска цена, да демонстрира силата на новите технологии в индустрията и да може да се използва така както в индустриална среда така и в домашни условия.

Едно такова решение ще може да приобщи хоби градинарите към иновативните технологии и ще направи гледането на растенията им по-лесно.

Пазарът за подобни устройства и софтуери е пълен. Обаче Botanica Wellness успява да се различи във всеки един аспект. Самите устройства които се използват за проследяването на растенията успяват да минамилизизарт разхода си на ток до минимум и в същото време да извлекат максималните функционалности от самите устройства.

## **Основна цел**

Основните цели на приложението и платформата могат да се разделят на три основни части, за трите основни групи на потребителите.

При домашните потребители, изградената платформа има за цел да улесни гледането на растенията в домашни условия и гледането им от разстояние. С помощта на функционалността за добавяне и задаване на системни и потребителски въведени референтни стойности за растенията (пресети), хоби градинарите могат дори да се запознаят повече с ботаниката като наука и да разберат как се отглеждат различните видове растения. Приложението е пригодено за всякакви потребителски групи и възрасти, заради интуитивния и удобен начин за проследяване на статистиките за растението в даден период от време и заради практичния му и изчистен интерфейс.

При индустриалното ползване на платформата и при селскостопанските производители основната цел е превенцията на заболявания по посевите, житните култури и селскостопанските растения за търговска употреба. Това съответно води и до останалите цели да се намалят разходите и загубите при такава епидемична обстановка при растенията. И едновременно с това се намалят и разходите по отглеждането на самите растения с устройствата на Botanica Wellness които работят в режин на ултра ниска консумация на ток.

И третата целева група са програмистите и разработчиците. Понеже проектът е напълно и изцяло с отворен код, всеки може да го изпробва и да прави негови собствени модифицирани версии (flavor-и). По този начин се стимулира общественото участие и сътрудничество, като позволява на хората от различни дисциплини да допринесат със своите знания и умения за развитието на софтуера. Също така предоставянето на приложение с отворен код може да има образователна ценност, като предоставя възможност за учене и обучение на студенти, учени и земеделски професионалисти за технологичните иновации в земеделието.

Системите с отворен код често са почва за иновации и експерименти, тъй като позволяват на разработчиците да изпробват нови идеи и функционалности, които могат да подобрят ефективността и резултатите в земеделието.

## **Очаквания**

Проектът Botanica Wellness очаква да постигне няколко ключови резултата. На първо място, той предвижда значително увеличение на иновациите и новите технологии в земеделието и тяхното разпространяване в сферата. Което от своя страна ще доведе до много различни резултати в различните потребителски групи.

Очакванията за проекта при домашните потребители е популяризирането не само на ботаниката като забавно хоби, но и запознаването на обикновения потребител с иновациите в индустрията и сферата на отглеждане на растения. За домашните градинари и любители на растенията, приложението може да предостави важна информация и статистика за ранно откриване и предотвратяване на болести по растенията, което ще помогне за поддържане на здравето и красотата на техните растения. Проектът предоставя лесен и интуитивен интерфейс, който домакините могат лесно да използват, без специални технически познания. Също така, мобилният клиент като част от платформата се очаква да бъде доразработен за различни видове устройство, като цели да достигне максимален брой хора. Очаква се също така значително спестяване на пари,време и усилия от страна на домакините които се грижат за растенията.

Очакванията в земеделската индустрия се препокриват малко или много с тези за домашните потребители. Качеството и количеството на различните реколти ще се повиши драстично с използването на софтуер за мониторинг на растения, базиран на IoT. Това включва и значителното намаляване на загубите на търговците причинени от тези дадени болести по растенията, което ще допринесе за по-устойчивото и икономически ефективно земеделие. Използването на технологията на IoT за мониторинг и контрол може да увеличи устойчивостта на земеделските производители към изменения в климата, екологични фактори и други предизвикателства, свързани със земеделието. Проектът също може да доведе до създаването на общност от земеделски производители, разработчици и експерти, които да споделят знания и опит, за да подобрят ефективността и резултатите в земеделието.

## **Предизвикателства**

При разработката на подобна система за мониторинг се очакват предизвикателства и проблеми зад всеки ъгъл. Предизвикателствата на проекта Botanica Wellness са от най-различни сфери. Като започнем от намирането на актуална тема за самия проект и стигнем до оптимизиране на потребителския интерфейс на приложението.

Първият проблем беше именно намирането на актуална тема и измислянето на практично и ефективно решения на дадения проблем. Това решение има за цел да разреши няколко проблема при няколко различни потребителски групи по един и същ унифициран начин като запази идентичността на приложението.

След това се появиха технологичните предизвикателства. Разработването на хардуерни и софтуерни компоненти за устройството изисква специфични технически умения, експертиза и познания. Освен това, осигуряването на съвместимост и стабилност между различните компоненти на системата само по себе си е предизвикателство.

При разработването на IoT устройствата се появи проблема за дизайна на самото устройство, какъв би бил най-подходящият хардуер за разработката му и как чрез персонализиран софтуер (firmware) за устройството да се намали консумацията му на ток. Други предизвикателства свързани с устройствата бяха начина по който те ще се свързват с BWS сървъра на приложението. Устройството трябва да бъде проектирано с мисъл за лесна употреба и привлекателен външен вид, за да бъде привлекателно за домашните потребители. Това включва съчетаване на функционалност с естетика и удобство за потребителите.

От друга страна най-сложната и комплексна част от проекта със сигурност е архитектурата на BWS сървъра. BWS представлява бекенд частта на цялата система, всички потребители, техните устройства и данните от устройствата. Чрез него се извършва също така цялостната комуникация между IoT устройствата и мобилния клиент на приложението.

# **Изложение**

## **Проучвателна част**

Целта на проекта е да създаде унифицирано решение за проследяване на състоянието на дадени растения и/или болестите които вървят по тях, без значение дали говорим за индустриално растениевъдство или професионално градинарство.

След проучване на темата и разучаване на подобни решения, се стигна до извода, че на пазара няма такава система която едновременно да може да работи за домашна употреба и индустриално растениевъдство и в същото време да е с отворен код и да има устройства с минимална консумация на ток и енергия.

Подобни софтуерни системи които работят на WSN принцип са **Edyn Garden Sensor, Parrot Flower Power(Фигура 1)** и др. Те се различават по това че са насочени само към любителите градинари, не са с отворен код и не се представят толко добре колкото **Botanica Wellness** като говорим за енергоразход на самите сензори и устройства.

Основните характеристики на проектът са, че интегрира Базирана на IoT система за мониторинг за контрол на епидемични заболявания, която е ключово приложение за прецизно земеделие за домашна и търговска употреба. Персонализиран софтуер за устройствата който позволява режим на ултра нисък енергоразход и поради това имат дълъг живот на батерията.

(Фигура 1)

## **Изисквания към програмния продукт**

### Поставени задачи

Задачите които бяха поставени в началото на проекта бяха сравнително прости и общи. Но с продължение на развиването му те се увеличаваха и ставаха все по детайлни. Основните етапи в създаването и разработка на проекта бяха:

* Разработване и модифициране на хардуерно устройство. Избор на подходящи технологии и проверка на съвместимостта им. Проектиране на компактен и удобен дизайн (Малък и дискретен дизайн, който може да се постави лесно в близост до растението без да заема много пространство).
* Подготовка на цялостната архитектура на приложението. Комуникацията между различните му компоненти.
* Разработка на персонализиран комуникационен протокол (API)
* Подготовка на физически сървър BWS и неговата настройка
* Подготовка на база данни PostgreSQL
* Планиране на необходимите функционалности
* Разработка на мобилен клиент
* Тестване в реална среда

Всички задачи от списъка са изпълнени успешно и проектът е функциониращ в първата му версия подготвена за публикуване.

* + 1. Бъдещо развитие

Бъдещето развитие на проекта се състои от няколко основни етапа**(Фигура 2)**. Първият етап е публикация на приложението глобално. Вторият е разработване на IOS и Web версия. Третият е поддръжка на повече от един вид устройство( различен софтуер за микроконтролерите на устройствата). И най-накрая интегриране на модул с изкуствен интелект за обработка на данните на дадените устройства и сензори за даване на подходящи съвети за отглеждане на растенията.

A diagram of a customization process

Description automatically generated

**(Фигура 2)**

## Същинска част на проекта

### IoT устройства

Устройствата използвани за проекта **(Фигура 3)** са базирани на готовия модул LILIGO Model T-Higrow. Те са модифицирани, за да може да отговарят на желанията на Botanica Wellness системата. Устройствата в тяхната същност представляват ESP32 микроконтролери с добавени сензори за температура, осветеност, влажност на въздуха и почвата и сензор за плодородност (минерализация) на почвата. Допълните сензори които се използват са DHT11(температура и влажност на въздуха), BH1750(осветеност) и WC сензор(за минерализация и влажност на почвата).

Устройствата работят с персонализиран софтуер проектиран и внедрен за хардуерните сензори за растения. Той е изграден като част от интегрираната система за мониторинг за наблюдение и прогнозиране на болести по растенията. Сензорите поддържат откриване на плодородието на почвата, откриване на влага в почвата, откриване на интензитета на светлината, температура на околния въздух, влажност на околния въздух, работен режим с ниска мощност, заспиване с ултра ниска мощност и поддръжка на дълъг живот на батерията. Сензорът за растенията се буди от режима си на дълбок сън, с който успява да пести батерия, на даден интервал зададен от потребителя при регистрация на новото устройство в профила му.

Тези хардуерни устройства работят на доста прост принцип, като повечето IoT устройства. При първоначалното пускане на устройството то ще работи в режим на SoftAP (Access Point). Това ще позволи на потребителя да се върже към локалната мрежа на устройството и да му даде нужната информация като потребителско име и парола на потребителя, SSID и парола на мрежата към която устройството ще седи вързано и дадения период през който устройството ще се буди за да вземе нужната информация от сензорите и да я изпрати към базата данни.

A plant with flowers in a pot

Description automatically generated

**(Фигура 3)**

### Мобилен Клиент

Приложението на Botanica Wellness**(Фигура 4)** е независим софтуерен модул за мобилни устройства с Android, изграден като част от интегрираната мониторингова система за наблюдение и прогнозиране на болести по растенията. Програмният продукт е разработен, за да улесни потребителите на системата, като им предостави мобилност и възможност за отдалечено конфигуриране, следене на данните от сензорите и известяване.

Приложението ще помогне на процеса на градинарство да се реализира по-ефективно, което увеличава приходите. Чрез събиране на различни параметри на данни към облачния сървър и активиране на дистанционно управление чрез мобилно приложение, базираната на IoT интелигентна система е доказано, че работи надеждно.

Софтуерът е проектиран и може:

* лесно да се инсталира и актуализира на най-разпространените мобилни устройства, базирани на операционна система Android
* чрез него да се създават, редактират или изтриват потребителски профили в BWS
* да се създават, редактират или изтриват неограничен брой профили на растения(пресети) с референтни стойности за най-добрите условия за тяхното отглеждане
* добавяне, редактиране или изтриване на неограничени безжични хардуерни сензори
* онлайн наблюдение и незабавно известяване (в процес на разработка), при несъответствие с дефинираните профили на растенията
* да бъдат разглеждани история и графични диаграми за дадените сензори

Използването на мобилно приложение е една от най-добрите практики за селскостопанска дейност, която използва най-малко време и помага за предотвратяване на увреждане на реколтата .Чрез приложение за Android земеделците и хоби градинарите могат да наблюдава онлайн целия процес на отглеждане.

A screenshot of a phone

Description automatically generated

**(Фигура 4)**

### BWS (Botanica Wellness Server)

Дизайнът на BWS е независим софтуерен модул, който не се влияе от типовете, броя и точността на хардуерните сензори, използвани в интелигентната система за наблюдение. Разработеният софтуер събира и анализира всички стойности от сензорите. Освен това софтуерът е гъвкав за използване с различни растения в наблюдавани полета и е проектиран да бъде независим от специфичен модел на заболяване. Планира се софтуерният модул да се състои от два компонента: компонент на системното ядро и компонент за изкуствен интелект (AI). Ядрото на системата се грижи за събирането на всички сензорни данни, а AI компонентът обработва и анализира показанията на сензора и изпраща аларми, ако е необходимо, за да защити посевите на потребителя. Botanica Cloud Server поддържа множество потребители и множество сензори на потребител. Към момента няма ограничение.

## **Описание на използвани софтуерни технологии**

### Уеб приложение

Уеб приложение, е софтуерно приложение, което е достъпно и управлявано изцяло чрез уеб браузър или по-точно казано през интернет. За разлика от традиционните софтуерни приложения, които изискват инсталиране на устройството на потребителя, уеб приложенията се хостват на отдалечени сървъри и се осъществяват достъп чрез уеб браузър, което ги прави независими от платформата и достъпни от всяко устройство с интернет връзка.

Уеб приложенията следват модел на архитектура клиент-сървър, където логиката и данните на приложението се намират на отдалечен сървър (backend), а потребителят взаимодейства с приложението чрез уеб браузър (клиент). Тази архитектура позволява безпроблемна комуникация между клиента и сървъра, като клиентът изпраща заявки до сървъра и получава отговори, съдържащи исканите данни или функционалност.

Потребителският интерфейс на уеб приложение се представя на потребителя чрез уеб браузър. Състои се от HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) и JavaScript, които заедно определят структурата, стила и интерактивността на интерфейса на приложението.

Съвременните уеб приложения често включват усъвършенствани UI framework-ове и библиотеки за създаване на динамични и отзивчиви потребителски интерфейси. Бекендът на уеб приложение се състои от код от страна на сървъра, отговорен за обработката на клиентски заявки, изпълнението на бизнес логиката и взаимодействието с бази данни или външни услуги.

Ролята на уеб приложение в случая на Botanica Wellness е web портала за регистрация на ново устройство. Самото приложение бива пуснато временно на уеб сървъра, който се пуска от самия микроконтролера. **(Фигура 5 и Фигура 6)**

A screenshot of a login screen

Description automatically generatedA screenshot of a wifi connection

Description automatically generated

**(Фигура 5 и Фигура 6)**

### Мобилно native приложение

Мобилно native приложение е софтуерна програма, разработена специално за конкретна мобилна операционна система, като Android или iOS, използваща „родните“ езици за програмиране, инструменти и библиотеки на платформата. За разлика от междуплатформените или хибридните приложения, които използват уеб технологии, native приложенията са изградени с помощта на специфични за платформата езици като Java или Kotlin за Android и Objective-C или Swift за iOS.. Android е по-разпространена от останалите операционни системи и предлага по-голям достъп до клиентска база. За това първата версия на мобилния клиент на Botanica Wellness е направен за Android.

Native приложенията предлагат висока производителност и безпроблемна интеграция с функции на устройството като камери, GPS и сензори, благодарение на достъпа до специфични за платформата API-ове. Те осигуряват удобно и интуитивно потребителско изживяване, като се придържат към насоките за дизайн на всяка платформа и стандартите за UI/UX, включително компоненти на собствения UI и модели за навигация.**(Фигура 7 и Фигура 8)**

Native приложенията също така често могат да осигурят стабилна офлайн поддръжка чрез кеширане на данни локално на устройството или използване на локална база данни, което позволява на потребителите да имат достъп до съдържание и да изпълняват задачи, дори когато са офлайн или имат ограничена връзка.

A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generatedРазпространени чрез официални магазини за приложения като Google Play Store и Apple App Store, native приложенията предлагат сигурност и доверие чрез прегледи и процеси на одобрение в магазина за приложения. Те също така поддържат стабилна офлайн функционалност и безпроблемно се интегрират с екосистемите на платформата за функции като насочени известия, покупки в приложението и удостоверяване. Въпреки че разработването на native приложения изисква експертни познания в специфични за платформата езици и инструменти, те осигуряват превъзходна производителност и потребителско изживяване, пригодени за всяка платформа.

**(Фигура 7 и Фигура 8)**

### Kotlin

Kotlin, модерен, статично въведен език за програмиране, придобива значителна популярност в разработката на Android поради своята оперативна съвместимост с Java и краткия, четим синтаксис. Той предлага функции като нулева безопасност, съпрограмми за асинхронно програмиране и разширения за Android за опростяване на разработката на потребителския интерфейс. Kotlin се поддържа официално от Google като първокласен език за разработка на Android, заедно с Java.

В допълнение към тези функции, Kotlin също се интегрира добре с Android Data Binding. Data Binding е библиотека, която позволява на разработчиците да обвързват UI компоненти в оформления с източници на данни в тяхното приложение, като използват декларативен формат. Това опростява кода за актуализиране на UI елементи въз основа на промени в основните данни, което води до по-поддържан и ефективен код. **(Фигура 9)**

Освен това Kotlin може безпроблемно да работи с WebView, който е компонент в Android, който позволява показване на уеб съдържание в приложение. Разработчиците могат да използват Kotlin за взаимодействие с екземпляри на WebView, зареждане на уеб страници, обработка на потребителски взаимодействия и изпълнение на JavaScript код, наред с други задачи. Тази гъвкавост позволява на разработчиците да създават богати на функции приложения, които безпроблемно интегрират уеб съдържание в родното Android изживяване. **(Фигура 9)**

Като цяло, комбинацията на Kotlin от модерни езикови функции, оперативна съвместимост с Java и безпроблемна интеграция с библиотеки на Android като Data Binding и WebView го правят мощен избор за създаване на приложения за Android. Нарастващото му приемане и силната подкрепа както от JetBrains, така и от Google продължават да го правят предпочитан език за разработка на Android.

****

**(Фигура 9)**

### WebView

WebView е основен компонент в разработването на Android, който позволява на разработчиците да вграждат уеб съдържание (HTML, CSS, JavaScript) в своите приложения за Android. Той предоставя начин за показване на уеб страници или уеб-базирано съдържание директно в потребителския интерфейс на приложението, позволявайки безпроблемна интеграция на уеб съдържание с родни изживявания на Android. WebView може да се използва за зареждане и показване на уеб страници от URL адреси, HTML файлове, съхранявани локално, или необработено HTML съдържание, предоставено програмно. Разработчиците могат да персонализират външния вид и поведението на WebView, като активиране/деактивиране на JavaScript, обработка на навигационни събития и контролиране на настройките на прозореца за изглед.

WebView позволява на разработчиците да обработват потребителски взаимодействия, като кликвания върху връзки или изпращане на формуляри, чрез различни обратни извиквания и слушатели на събития. Това позволява на разработчиците да осигурят богато изживяване при сърфиране в рамките на приложението и да внедрят персонализирани поведения въз основа на взаимодействията на потребителите с уеб съдържанието. WebView предоставя API за изпълнение на JavaScript код в заредените уеб страници. Това позволява на разработчиците да взаимодействат динамично с уеб съдържанието, да манипулират DOM (Document Object Model) и да обменят данни между WebView и уеб страницата, използвайки JavaScript интерфейси.

Разработчиците могат да зареждат както локално, така и отдалечено уеб съдържание в копия на WebView. Локалното съдържание може да се съхранява в папката с активи или ресурси на приложението, докато отдалеченото съдържание се зарежда от URL адреси, посочени от разработчика, или се получава динамично от сървър. WebView предлага различни опции за персонализиране и настройки за конфигурация, за да контролира поведението и външния вид. Разработчиците могат да конфигурират настройки, свързани с кеширане, бисквитки, мащабиране и други, за да оптимизират производителността на WebView и потребителското изживяване въз основа на изискванията на приложението.

Тъй като WebView изобразява уеб съдържание в контекста на приложението, са взети предвид последиците за сигурността, като защита срещу злонамерено съдържание, предотвратяване на атаки със скриптове между сайтове (XSS) и осигуряване на сигурна комуникация с външни сървъри при зареждане на отдалечено съдържание. Като цяло WebView е многофункционален инструмент за интегриране на уеб съдържание в приложения за Android, което позволява на разработчиците да създават хибридни приложения, които безпроблемно съчетават собствени и уеб базирани функции. Използвайки WebView с Kotlin, разработчиците могат да създадат мощни и богати на функции приложения, които осигуряват сплотено потребителско изживяване както в естествена, така и в уеб среда. Botanica Wellness работи с WebView понеже ще се интегрира сравнително по-бързо на останалите платформи и ще запази еднаква функционалност на различните устройства.

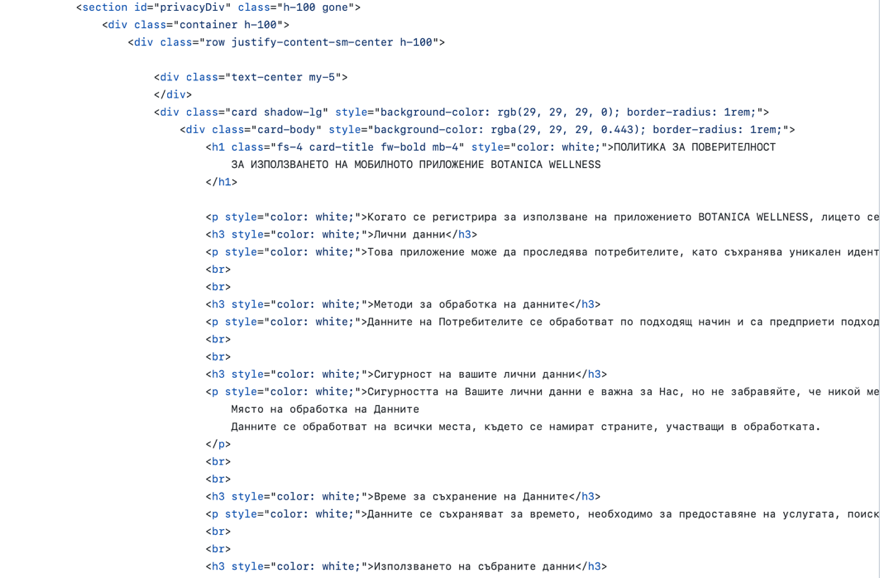
### HTML

HTML (HyperText Markup Language) служи като гръбнакът на световната мрежа, осигурявайки структурата и съдържанието за почти всяка уеб страница, която срещнете онлайн. Това е език за маркиране, който дефинира структурата на уеб документи чрез използване на система от тагове и атрибути за описание на елементите в една страница. В основата си HTML се състои от набор от таг-ове за маркиране, които определят структурата и съдържанието на уеб страница.

Чрез тях се форматира, графично оформя текста и неговите отделните части в рамките на една еб страница, като например заглавия, цитати, текстови раздели, хипертекстови препратки и т.н. Тези тагове са затворени в ъглови скоби (< >) и обикновено идват по двойки - отварящ таг и затварящ таг. Отварящият таг показва началото на елемент, докато затварящият таг обозначава края на този елемент. Някои елементи обаче са само затварящи се и не изискват затварящ таг. **(Фигура 10)**

HTML елементите могат да бъдат категоризирани в няколко типа: семантични елементи, елементи на текстово съдържание, мултимедийни елементи, елементи на формуляри и елементи на структурата на документа.

HTML документите обикновено са структурирани йерархично, с вложени елементи, образуващи дървовидна структура.

Тази йерархична структура е от съществено значение за организиране и стилизиране на съдържанието на уеб страница с помощта на CSS (Cascading Style Sheets) и за добавяне на интерактивност и функционалност с помощта на JavaScript.

**(Фигура 10)**

### CSS

CSS или Cascading Style Sheets е основна технология в уеб разработката, която допълва HTML чрез контролиране на представянето и оформлението на уеб документите. Той позволява на разработчиците да стилизират и проектират уеб страници, осигурявайки последователност, естетика и приятно потребителско изживяване на различни устройства и размери на екрана. В основата си CSS се състои от набори от правила, които определят как HTML елементите трябва да се показват в браузъра. Тези набори от правила се състоят от селектори, свойства и стойности. Селекторите са насочени към конкретни HTML елементи, докато свойствата определят визуалните характеристики на тези елементи, като цвят, размер, шрифт, разстояние и позициониране. **(Фигура 11)**

Стойностите определят специфичните настройки за всяко свойство. CSS предоставя широк набор от възможности за стилизиране на уеб съдържание, включително селектори, модел на кутия, типография, цветове и фонове, оформление, преходи и анимации и медийни заявки.

A screenshot of a computer code

Description automatically generatedВ обобщение, CSS е критична технология в уеб разработката, която позволява на разработчиците да стилизират и проектират уеб страници, да контролират оформлението и типографията, да прилагат цветове и фонове, да създават адаптивни оформления, да добавят преходи и анимации и да оптимизират потребителското изживяване на различни устройства и размери на екрана. Използвайки силата на CSS, разработчиците могат да създават визуално привлекателни и ангажиращи уеб приложения, които завладяват потребителите и ефективно придават съдържание и идентичност на сайта.

**(Фигура 11)**

### JavaScript

JavaScript е многофункционален език за програмиране, който обикновено се използва за създаване на интерактивно и динамично уеб съдържание. Като основен компонент на уеб разработката, JavaScript позволява на разработчиците да добавят функционалност, да манипулират данни и да реагират на потребителските взаимодействия в уеб приложенията. JavaScript се използва предимно за скриптове от страна на клиента, което означава, че работи в уеб браузъра на потребителя, а не на отдалечен сървър. Това позволява на JavaScript да взаимодейства с Document Object Model (DOM), който представлява структурата на HTML и XML документи. Чрез достъп до и модифициране на DOM, JavaScript може динамично да актуализира съдържанието и поведението на уеб страниците в отговор на потребителски действия, като кликвания, събития за въвеждане и събития за зареждане на страници.

Ключовите характеристики и възможности на JavaScript включват променливи и типове данни, оператори и изрази, оператори на потока на управление, функции, обекти и обектно-ориентирано програмиране (ООП), обработка на събития, асинхронно програмиране, обработка на грешки, модули и модуларизация и API на браузъра.

A computer screen shot of a program

Description automatically generated

**(Фигура 12)**

### Apexcharts

ApexCharts е JavaScript библиотека с отворен код, която позволява на разработчиците да създават интерактивни и визуално привлекателни диаграми за уеб приложения. В Botanica Wellness, библиотеката играе ключова роля във визуализирането на дадена статистика от данни от IoT устройствата за период от време. Създаден върху популярната библиотека с диаграми ApexCharts.js, той предоставя широка гама от типове диаграми, включително линейни диаграми, диаграми с площи, стълбовидни диаграми, колонни диаграми, точкови диаграми, кръгови диаграми, кръгови диаграми, радарни диаграми и др. Тези диаграми са силно интерактивни и динамични, позволявайки на потребителите да задържат курсора на мишката върху точки с данни, за да видят подсказки, да увеличават и намаляват мащаба за подробен анализ и да превключват видимостта на серията за фокусиране върху конкретни набори от данни.

Една от ключовите характеристики на ApexCharts е неговата отзивчивост, която гарантира, че диаграмите се адаптират безпроблемно към различни размери на екрана и устройства. Освен това разработчиците разполагат с обширни опции за персонализиране, включително цветове, шрифтове, етикети, подсказки, линии на мрежата и легенди, което им позволява да приведат диаграмите в съответствие с изискванията за марка и стил на тяхното приложение. Библиотеката също така поддържа анимации, осигурявайки плавни преходи и визуални сигнали по време на изобразяване или актуализации на данни.

ApexCharts може да обработва както статични, така и динамични източници на данни, позволявайки на разработчиците да предоставят данни директно или динамично да ги актуализират в реално време с помощта на JavaScript. Освен това библиотеката дава приоритет на стандартите за достъпност, като гарантира използваемост от хора с увреждания чрез поддръжка за навигация с клавиатура, екранни четци и други функции за достъпност.

Като цяло ApexCharts е мощен и многофункционален инструмент за създаване на зашеметяващи визуализации на данни в уеб приложения. Неговият богат набор от функции, опции за персонализиране, отзивчивост и поддръжка за достъпност го правят популярен избор сред разработчиците, които искат да подобрят представянето на своите данни.

### PHP

PHP (Hypertext Preprocessor) е скриптов език от страна на сървъра, широко използван за уеб разработка. Първоначално създаден от датско-канадския програмист Расмус Лердорф през 1994 г., PHP оттогава се превърна в един от най-популярните езици за програмиране за изграждане на динамични и интерактивни уеб приложения.

PHP се използва предимно за скриптове от страна на сървъра, което означава, че се изпълнява на уеб сървъра, преди полученият HTML да бъде изпратен до уеб браузъра на клиента. Това позволява на PHP да генерира динамично съдържание, да взаимодейства с бази данни, да обработва подавания на формуляри и да изпълнява други задачи от страна на сървъра.

Като отворен код, PHP е свободно достъпен за използване, модифициране и разпространение от всеки, което допринася за широкото му възприемане и развитието на голяма и активна общност от разработчици. Неговата простота и лекота на използване го правят достъпен както за начинаещи, така и за опитни разработчици, със синтаксис, подобен на C и Perl.

PHP е независим от платформата и е съвместим с повечето уеб сървъри, което му позволява да работи на различни операционни системи като Linux, Windows, macOS и Unix. Той се интегрира безпроблемно с HTML, позволявайки на разработчиците да смесват PHP и HTML в един и същи документ, за да създават динамични уеб страници.

С вградена поддръжка за взаимодействие с бази данни като MySQL, PostgreSQL и SQLite, PHP е много подходящ за изграждане на уеб приложения, управлявани от бази данни. Той също така предлага богат набор от вградени функции и библиотеки за задачи като обработка на файлове, манипулиране на низове, манипулиране на дата и час и обработка на изображения.

PHP има голяма и активна общност от разработчици, които допринасят за разработването на рамки, библиотеки и инструменти, които разширяват неговата функционалност и опростяват задачите за уеб разработка. Популярните PHP рамки включват Laravel, Symfony, CodeIgniter и Yii, осигуряващи структуриран подход за изграждане на уеб приложения.

В обобщение, PHP е мощен и многофункционален скриптов език от страна на сървъра, който остава популярен избор за уеб разработка поради своята лекота на използване, независимост от платформата, интегриране на бази данни, широка функционалност и силна подкрепа от общността.

### Custom API

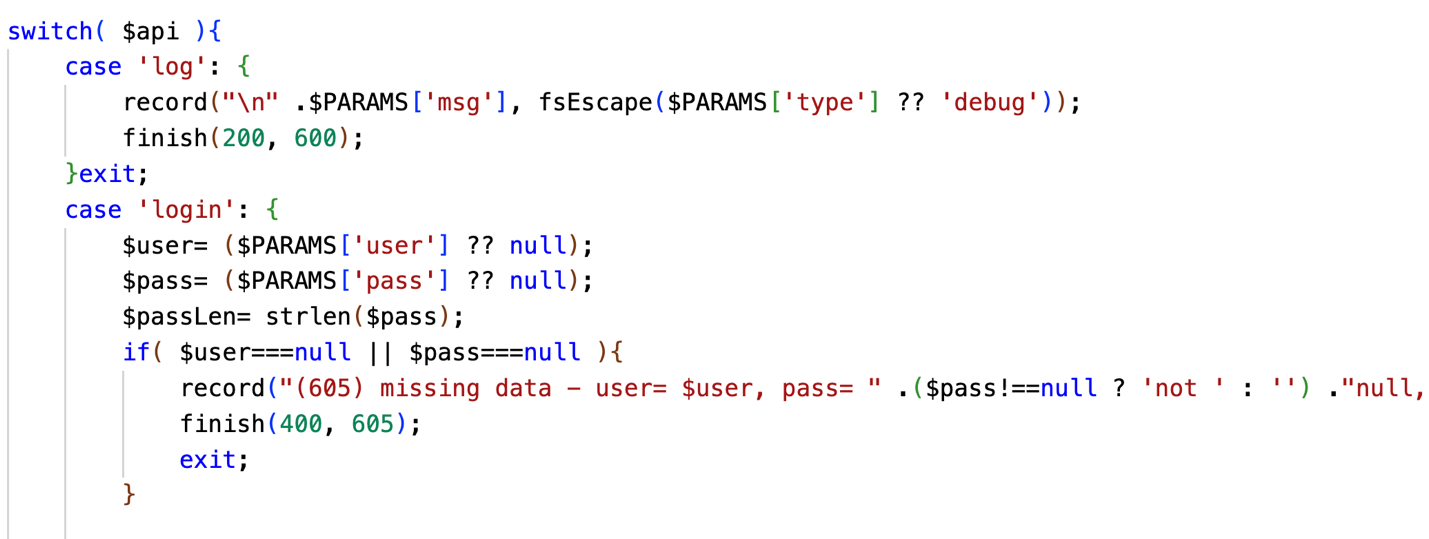
Комуникацията между различните компоненти на системата се случва именно посредством персонализиран API написан на PHP. Чрез този API при изпращане на HTTPS заявка от приложението към BWS или от устройство към BWS в хедърите на заявката се подава параметър “api”, който има различни стойности стойности в зависимост от желаната от нас заявка. Например една заявка за влизане в потребителски профил от приложението ще изглежда по този начин**.(Фигура 13)**

**A screen shot of a computer code

Description automatically generated**

**(Фигура 13)**

А обработка от страна на BWS бекедна ще изглежда по този начин.**(Фигура 14)**

****

**(Фигура 14)**

### PostgreSQL

PostgreSQL, често наричан "Postgres", е мощна система за управление на релационни бази данни (RDBMS) с отворен код, известна със своята надеждност, устойчивост и разширени функции. Разработен от PostgreSQL Global Development Group, той организира данните в таблици, състоящи се от редове и колони, установявайки връзки между таблиците, използвайки първични и външни ключове. PostgreSQL е съвместим с ACID (атомарност, съгласуваност, изолация, издръжливост), като гарантира надеждно и безопасно изпълнение на транзакции с база данни за целостта и последователността на данните.

Като отворен код, PostgreSQL позволява на потребителите свободно да използват, модифицират и разпространяват софтуера, насърчавайки сътрудничеството и развитието, управлявано от общността. Той предлага широк набор от разширени функции, включително поддръжка за сложни типове данни, възможности за търсене в пълен текст, поддръжка на геопространствени данни и дефинирани от потребителя функции (UDF). Освен това PostgreSQL поддържа усъвършенствани техники за индексиране и методи за оптимизация, за да осигури висока производителност и производителност за взискателни работни натоварвания.

PostgreSQL е силно разширяем и адаптивен, което позволява на потребителите да създават потребителски типове данни, функции и разширения, за да отговорят на специфични изисквания. Той поддържа множество процедурни езици за писане на съхранени процедури и тригери, като подобрява гъвкавостта при разработката. Сигурността е с приоритет в PostgreSQL с функции като ролеви контрол на достъпа (RBAC), SSL криптиране и усъвършенствани механизми за удостоверяване за защита на чувствителни данни.

Проектиран да мащабира вертикално и хоризонтално, PostgreSQL поддържа големи обеми от данни и високи скорости на транзакции, което го прави подходящ за приложения с критично значение. Той е съвместим с различни платформи, работи на различни операционни системи и предоставя клиентски библиотеки и драйвери за безпроблемна интеграция с популярни езици за програмиране. Като цяло, PostgreSQL е зряла, богата на функции система от бази данни, на която организациите имат доверие за изграждане на приложения и услуги, управлявани от данни.

### NGINX

Nginx (произнася се "engine-x") е високопроизводителен уеб сървър с отворен код, обратен прокси сървър и балансьор на натоварването, известен със своята скорост, ефективност и мащабируемост и е основата на която седи BWS. Разработен през 2004 г., за да се справи с ограниченията на традиционните уеб сървъри, Nginx получи широко разпространение и се използва от милиони уебсайтове и уеб приложения по целия свят.

Nginx е известен със своята висока производителност и ниско потребление на ресурси, което го прави идеален за работа с голям обем трафик и бързо доставяне на съдържание на потребителите. Неговата управлявана от събития архитектура и асинхронен, неблокиращ I/O модел му позволяват да обработва хиляди едновременни връзки ефективно.

Като уеб сървър, Nginx обслужва статично съдържание, като HTML, CSS, JavaScript и медийни файлове, както и динамично съдържание, генерирано от приложения, работещи на бекенд сървъри. Той поддържа виртуален хостинг, SSL/TLS криптиране, пренаписване на URL адреси и други функции, необходими за хостване на уебсайтове и уеб приложения.

Освен това Nginx може да действа като обратен прокси сървър, като седи пред задните сървъри и препраща клиентски заявки към тях. Това позволява балансиране на натоварването, кеширане и маршрутизиране на заявки, подобрявайки производителността, надеждността и скалируемостта на уеб приложенията. Той може също така да функционира като балансьор на натоварването, разпределяйки входящите заявки между множество бекенд сървъри, за да осигури оптимално използване на ресурсите и да предотврати претоварване на отделни сървъри.

Nginx включва вградени възможности за кеширане за съхраняване и обслужване на често достъпно съдържание от паметта, намалявайки натоварването на бекенд сървърите и подобрявайки времето за реакция за потребителите. Той поддържа кеширане както на статично, така и на динамично съдържание, с конфигурируеми правила за кеширане и времена на изтичане.

Освен това Nginx често се използва в архитектурите на микроуслуги като обратен прокси и балансьор на натоварването, маршрутизирайки заявки към различни микроуслуги въз основа на URL пътища или други критерии. Той също така поддържа модерни уеб протоколи като HTTP/2 и осигурява стабилни възможности за TLS/SSL криптиране за осигуряване на комуникация между клиенти и сървъри.

Nginx е силно разширяем, с богата екосистема от модули и плъгини на трети страни, налични за добавяне на допълнителна функционалност и персонализиране. Разработчиците могат да пишат персонализирани модули на C или да използват съществуващи модули, за да разширят възможностите на Nginx според техните специфични изисквания.

Като цяло Nginx е многофункционален и мощен компонент на съвременната уеб инфраструктура, използван за подобряване на производителността, надеждността и скалируемостта на уеб приложенията. Лекият му отпечатък, гъвкавостта и разширените функции го правят популярен избор за организации от всякакъв размер и индустрии.

### Софтуер за IoT устройства

Софтуерът или така нареченият фирмуер на IoT устройствата е базиран на популярния език за програмиране на микроконтролери, който е вариация на C++. **(Фигура 15)**

Езикът за програмиране Arduino е опростена версия на езика за програмиране C++. Той е специално проектиран да улесни начинаещите да започнат с програмирането на микроконтролери като тези, които се намират на платките Arduino.

Програмният език Arduino е подобен на C++ по много начини, но включва редица вградени функции и библиотеки, които улесняват взаимодействието с хардуера на платките. Например, има вградени функции за управление на цифрови и аналогови пинове, сензори за четене и комуникация с други устройства.

Ето някои от основните характеристики на езика за програмиране Arduino:

Setup() и Loop(): Всяка скица на Arduino има две основни функции: setup() и loop(). Функцията setup() се извиква веднъж, когато платката се включи или нулира, и се използва за инициализиране на променливи, задаване на режими на щифтове и конфигуриране на платката. Функцията loop() се извиква многократно, докато платката работи, и се използва за контролиране на основното поведение на платката.

Режими на щифта: В езика за програмиране Arduino можете да зададете режима на щифта на INPUT или OUTPUT. Това ви позволява да контролирате дали щифтът се използва за четене на данни от сензор или за изпращане на данни към друго устройство.

Цифрови и аналогови I/O: Езикът за програмиране Arduino включва вградени функции за управление на цифрови и аналогови пинове на платката. Това позволява да се контролират светодиоди, двигатели и други устройства, които изискват цифрови или аналогови сигнали.

Библиотеки: Езикът за програмиране Arduino включва редица вградени библиотеки, които могат да се използват за опростяване на общи задачи, като управление на LCD дисплеи, сензори за четене и комуникация с други устройства.

Като цяло езикът за програмиране Arduino е проектиран да бъде лесен за научаване и използване, дори и за хора без опит в програмирането. Той позволява да взаимодействате с хардуера на платките по прост и интуитивен начин, което го прави чудесен избор за създаване на прототипи и експериментиране с електронни проекти.

A screenshot of a computer program

Description automatically generatedСофтуерът на IoT устройствата на Botanica Wellness има за цел да накара устройствата да изпращат нужната информация към BWS за съхранение и визуализиране от страна на приложението.

**(Фигура 15)**

## **Описание на използвани хардуерни технологии**

### ESP32 Chipset

Чипсетът ESP32, разработен от Espressif Systems, представлява мощна и гъвкава система върху чип (SoC), широко използвана в Интернет на нещата (IoT) и приложения за вградени системи. С двуядрен микропроцесор Xtensa LX6, работещ със скорости до 240 MHz, ESP32 улеснява възможностите за многозадачност, подобрявайки цялостната производителност и отзивчивост.

За отбелязване е, че ESP32 е оборудван с вграден Wi-Fi (802.11 b/g/n) и Bluetooth свързаност, което го прави подходящ за безброй IoT приложения, изискващи безжична комуникация. Нещо повече, неговата поддръжка за технологията Bluetooth Low Energy (BLE) позволява на устройствата да комуникират безжично на къси разстояния с минимална консумация на енергия, често използвана в IoT приложения като наблюдение на сензори и дистанционно управление.

Чипсетът предлага богат набор от периферни устройства, включително GPIO пинове, SPI, I2C, UART, ADC, DAC и други, улесняващи гъвкавото взаимодействие със сензори, задвижващи механизми, дисплеи и други външни устройства. Енергийната ефективност е отличителна черта на ESP32, благодарение на неговия дизайн с изключително ниска консумация на енергия. С множество режими на захранване, включително режим на дълбок сън, разработчиците могат да оптимизират потреблението на енергия въз основа на изискванията на приложението, което го прави много подходящ за IoT устройства, захранвани от батерии.

Функциите за сигурност са неразделна част от ESP32, с хардуерно базирани мерки като сигурно стартиране, криптографско ускорение, хардуерно генериране на произволни числа и сигурно съхранение за ключове и сертификати. Тези функции укрепват сигурността на данните и комуникациите в IoT приложенията.

Espressif предоставя стабилна екосистема за разработка за ESP32, включително комплекти за разработка на софтуер (SDK), библиотеки, документация и инструменти. Разработчиците могат да избират от различни платформи за програмиране като Arduino IDE, ESP-IDF (IoT Development Framework) или други популярни среди за разработка.

И накрая, чипсетът ESP32 е известен със своята достъпност, наличност и гъвкавост. Неговата комбинация от характеристики, производителност и рентабилност го направи предпочитан избор сред любители, създатели и професионални разработчици за създаване на прототипи и внедряване на IoT решения.

В обобщение, чипсетът ESP32 представлява всеобхватна и способна платформа за разработване на IoT и приложения за вградени системи, предлагайки безжична свързаност, богати периферни устройства, ниска консумация на енергия, функции за сигурност и обширна поддръжка за разработка.

### DHT11

Сензорът DHT11 е популярен и широко използван цифров сензор за температура и влажност в общностите на производителите и IoT. Разработен от компанията Aosong, сензорът DHT11 е известен със своята простота, достъпност и лекота на използване. Той е способен да измерва както температурата, така и нивата на влажност в околната среда, като предоставя цифрови изходни сигнали, представящи показанията. Сензорът комуникира с помощта на едножилен цифров протокол, опростявайки процеса на свързване и свързване. Въпреки ниската си цена, сензорът DHT11 осигурява надеждни измервания на температура и влажност, подходящи за различни приложения. Неговата точност е умерена, като точността на температурата обикновено е в рамките на ±2°C и точността на влажността в рамките на ±5%. Сензорът работи в широк температурен диапазон от 0°C до 50°C и нива на влажност от 20% до 90%. С ниска консумация на енергия, той е подходящ за захранвани с батерии и енергийно ефективни приложения. Сензорът DHT11 е широко достъпен от различни доставчици и е съвместим с популярни микроконтролери и платформи за разработка, което го прави лесен за интегриране в проекти и прототипи. Намира приложения в мониторинга на времето, наблюдението на околната среда, домашната автоматизация, ОВК системите, селското стопанство и индустриалния мониторинг.

### BH1750 sensor

Сензорът BH1750, известен още като BH1750FVI, е цифров сензор за околна светлина, предназначен да измерва интензитета на околната светлина в различни приложения. Разработен от ROHM Semiconductor, сензорът BH1750 е високо ценен заради своята точност, лекота на използване и гъвкавост. Той използва фотодиоден масив, за да открие интензитета на светлината, падаща върху повърхността му, и преобразува това измерване в цифрови изходни сигнали. С широк диапазон на измерване, обикновено вариращ от 0 до 65 535 лукса и висока точност до разделителна способност от 1 лукс, той позволява прецизно наблюдение и контрол на условията на осветление в различни среди. Сензорът BH1750 комуникира с външни устройства, използвайки прост цифров интерфейс, обикновено чрез протокола I2C, което го прави лесен за взаимодействие с микроконтролери, Arduino платки, Raspberry Pi и други цифрови устройства. Отличаващ се с ниска консумация на енергия и компактен пакет за повърхностен монтаж, той е подходящ за захранвани с батерии и енергийно ефективни приложения, с минимално въздействие върху общия размер и тегло на крайния продукт. Сензорът BH1750 намира приложения в интелигентни системи за осветление, автоматичен контрол на яркостта за дисплеи, системи за управление на енергията, IoT устройства, носима електроника и системи за мониторинг на околната среда. Като цяло, това е надеждно и универсално решение за измерване на интензитета на околната светлина в различни приложения.

### Dell PowerEdge R720

Dell PowerEdge R720 е многофункционален сървър, проектиран за широк спектър от корпоративни компютърни нужди. Издаден от Dell като част от своята линия PowerEdge, R720 предлага висока производителност, мащабируемост и надеждност, което го прави подходящ за различни приложения, включително виртуализация, центрове за данни, облачни изчисления и високопроизводителни изчисления (HPC). Сървърът поддържа разнообразие от процесори от серията Intel Xeon E5-2600, осигурявайки до 24 ядра и 48 нишки на сървър за високи нива на производителност и ефективност. С голям капацитет на паметта до 24 DIMM слота и максимален капацитет на паметта от 768GB (с RDIMM) или 1,5TB (с LRDIMM), R720 побира приложения с интензивна памет и виртуализирани среди. Той предлага гъвкави опции за съхранение, поддържайки до шестнадесет 2,5-инчови или осем 3,5-инчови SAS, SATA или SSD устройства с горещо включване, с интегрирани RAID контролери за защита на данните и резервиране. Сървърът предоставя широки възможности за разширение с множество PCIe разширителни слота, включително PCIe 3.0 слотове за високоскоростна връзка и допълнителни опции за съхранение или мрежови връзки. Опциите за работа в мрежа включват интегрирани Gigabit Ethernet портове като стандарт, с опционални надстройки до 10GbE и InfiniBand свързаност. Интегрираните инструменти за управление на Dell, като iDRAC (интегриран контролер за отдалечен достъп на Dell) с контролер за жизнен цикъл, предоставят цялостни възможности за управление и наблюдение за дистанционно управление, актуализации на фърмуера и системна диагностика. Излишните захранвания, вентилаторите с възможност за гореща смяна и поддръжката на ECC памет гарантират надеждност и непрекъсната работа. Със своята висока производителност, капацитет на паметта и гъвкави опции за съхранение, R720 е много подходящ за виртуализация и облачни изчислителни среди. R720 e сърцето на BWS, сървърът е предвиден да може да издържи на високи натоварвания от много потребители и от много устройства едновременно.

## **Ръководство на потребителя**

* + 1. Ръководство за краен потребител
* Активиране/Зареждане на устройството
* Регистрация на потребителски профил през мобилното приложение
* Влизане в уеб портала на устройството и попълване на нужните данни
* Поставяне на устройството в саксия или в близост до растение
* Настройване на пресети и референтни стойности(при желание на потребителя)
  + 1. Ръководство за разработчик

**Системни изисквания за сървъра**:

Хардуер:

Памет - Botanica Wellness Server изисква както физическа, така и дискова памет. Обемът на необходимата дискова памет очевидно зависи от броя на хостовете и параметрите, които се наблюдават. Ако планирате да поддържате дълга история на наблюдаваните параметри, трябва да помислите за поне няколко гигабайта, за да имате достатъчно място за съхраняване на историята в базата данни.

Процесор - BWS и особено базата данни Botanica Wellness може да изискват значителни процесорни ресурси в зависимост от броя на наблюдаваните параметри.

Примери за хардуерна конфигурация:

Таблицата предоставя примери за хардуерна конфигурация**.(Фигура 16)**

Това са примери за размер и хардуерна конфигурация, с които да започнете. Всяка BWS инсталация е уникална. Уверете се, че сте сравнили производителността на вашата система Botanica Wellness в среда за етап или разработка, така че да можете да разберете напълно вашите изисквания, преди да разположите BWS инсталацията в нейната производствена среда.

A screenshot of a white table

Description automatically generated

**(Фигура 16)**

Пример с екземпляри EC2 с общо предназначение на Amazon, използващи архитектура ARM64 или x86\_64, подходящ тип екземпляр като Compute/Memory/Storage optimized трябва да бъде избран по време на оценката и тестването на инсталацията на BWS, преди да се инсталира в неговата производствена среда.

Необходим софтуер:

BWS е изграден около модерни уеб сървъри, водещи машини за бази данни и PHP скриптов език.

Външен софтуер нужен за инсталацията: A screenshot of a computer

Description automatically generated

**(Фигура 17)**

Времева синхронизация:

Много е важно да имате точно системно време на сървъра с работещ BWS. „ntpd“ е най-популярният даемон, който синхронизира времето на хоста с времето на други машини. Силно се препоръчва да поддържате синхронизирано системно време на всички системи, на които работят компонентите на Botanica Wellness.

Мрежови изисквания:

A screenshot of a phone

Description automatically generatedСледният списък с отворени портове за компонент е приложим за конфигурация по подразбиране.

**(Фигура 18)**

**Настройване на среда за разработка и инсталиране на фирмуер на устройството:**

1. Инсталирайте разширение PlatformIO: Отворете VS Code и отидете до изгледа Extensions. Потърсете „PlatformIO IDE“ и го инсталирайте.
2. Създайте нов проект: Във VS Code отворете командната палета (Ctrl+Shift+P на Windows/Linux, Cmd+Shift+P на macOS) и изберете „PlatformIO: Нов проект“. Следвайте подканите, за да настроите вашия проект, включително избор на платка (по ваш избор от поддържаните платки) и рамка (Arduino, ESP-IDF и др.).
3. Вземете кода: Намерете поддържания от вашето устройство фирмуер на „botanica-sensors-firmware/името на вашето устройство“.
4. Конфигурация на PlatformIO: Прегледайте файла platformio.ini в директорията на вашия проект. Този файл съдържа конфигурационни настройки, като целева платка, скорост на качване и допълнителни библиотеки. Уверете се, че настройките отговарят на вашите изисквания за модул/устройство и проект.
5. Изграждане и качване: Преди да качите каквото и да е, ще трябва да изградите изображение на файловата система и след това да качите това изображение на устройството. Използвайте лентата с инструменти PlatformIO във VS Code, за да създадете и качите изображението на вашата файлова система в платката ESP32. Кликнете върху съответните икони, за да компилирате и качите фърмуера. Като алтернатива можете да използвате командната палета и да изберете команди PlatformIO.

**Настройване на среда за разработка и инсталиране на мобилен клиент:**

**0. Общи софтуерни зависимости.**

Необходими са някои инструменти, без значение каква е вашата конкретна IDE или инструмент за изграждане на предпочитания.

git се използва за достъп до различните версии на изходния код на Botanica Wellness. Изтеглете и инсталирайте версията, подходяща за вашата операционна система от тук. Добавете пълния път към директорията „bin/“ от вашата git инсталация в променливата PATH на вашата среда, така че да може да се използва от всяко място.

SDK за Android е необходим за изграждане на приложението. Има различни опции за инсталиране във вашата система в зависимост от IDE, която решите да използвате. Проверете документацията на Google относно инсталирането за повече подробности относно тези опции. След като го инсталирате, добавете пълния път към директориите „tools/“ и „platform-tools/“ от вашата инсталация на Android SDK в променливата PATH на вашата среда.

Отворете терминал и въведете „android“, за да стартирате Android SDK Manager. За да изградите приложението Botanica Wellness за Android, ще трябва да инсталирате поне следните SDK пакети:

Android SDK Tools и Android SDK Platform-tools (вече инсталирани); надграждането до последните им версии обикновено е добра идея.

Вече няма нужда да посочвате версия за инструментите за изграждане, приставката Gradle използва минималната необходима версия по подразбиране.

Android 12.0 (API 31), SDK платформа; необходими за изграждане на приложението Botanica Wellness.

Инсталирайте всеки друг пакет, който смятате за интересен, като например емулатори.

За други софтуерни зависимости проверете подробностите в раздела, съответстващ на предпочитаната от вас IDE или система за компилация.

**1. Направете fork и изтеглете Flowers-At-Home-2.0/botanica-mobile-android-client repository.**

Ще ви трябва git за достъп до различните версии на изходния код на Botanica Wellness. Изходният код се хоства в Github и може да бъде прочетен от всеки, без да е необходим акаунт в Github. Ще ви е необходим акаунт в Github, ако искате да допринесете за развитието на приложението със собствен код.

Следващите стъпки предполагат, че имате акаунт в Github и че ще получите кода от собствения си форк.

В уеб браузър отидете на връзката за репо и щракнете върху бутона „Форк“ близо до горния десен ъгъл.

Отворете терминал и продължете със следващите стъпки в него.

Клонирайте вашето разклонено хранилище: git clone https://github.com/YOURGITHUBNAME/Flowers-At-Home-2.0.git.

Преместете в папката на проекта с CD Flowers-At-Home-2.0.

Извлечете и приложете всички промени от вашия отдалечен клон 'master': git fetch + git rebase

Направете официално Botanica Wellness репозитори, известно като upstream: git remote add upstream https://github.com/Flowers-At-Home-2.0/android.git

Уверете се, че получавате и прилагате най-новите промени от официалния клон на android/master: git fetch upstream + git rebase upstream/master

На този етап можете да продължите да използвате различни инструменти за изграждане на проекта. Раздели 2 и 3 описват съществуващите алтернативи.

2. Работа с Android Studio.

В момента Android Studio е официалната IDE за Android. Поради това, ние го препоръчваме като IDE за използване във вашата среда за разработка.

Препоръчваме да използвате последната налична версия в стабилния канал за актуализации на Android Studio. Вижте какъв канал за актуализация проверява вашето Android Studio за актуализации в пътя на менюто „Помощ“/„Проверка за актуализация...“/връзка „Актуализации“ в диалоговия прозорец.

За да настроите проекта в Android Studio, изпълнете следните стъпки:

Отворете Android Studio и изберете „Импортиране на проект (Eclipse ADT, Gradle и т.н.)“. Прегледайте вашата файлова система до папката „android“, където се намира проектът. След това Android Studio ще създаде необходимите „.iml“ файлове. Ако някога затворите проекта, но файловете са все още там, просто изберете „Отвори проект...“. Инструментът за избор на файл ще покаже лице на Android като икона на папка, която можете да изберете, за да отворите отново проекта.

Android Studio ще се опита да изгради проекта директно след импортирането му. За да го изградите ръчно, следвайте пътя на менюто „Build“/„Make Project“ или просто щракнете върху бутона „Play“ в лентата с инструменти, за да го изградите и стартирате в мобилно устройство или емулатор. Полученият APK файл ще бъде записан в поддиректорията „build/outputs/apk/“ в папката на проекта.

3. Работа в терминал с Gradle:

Gradle е системата за изграждане, използвана от Android Studio за управление на операциите за изграждане на приложения за Android. Не е необходимо да инсталирате Gradle във вашата система и Google препоръчва да не го правите, а вместо това да се доверите на обвивката на Gradle, включена в проекта.

Отворете терминал и отидете в директорията „android“, която съдържа хранилището.

Стартирайте задачите „почистване“ и „изграждане“, като използвате предоставената обвивка на Gradle

Windows: gradlew.bat чиста компилация

Mac OS/Linux: ./gradlew чиста компилация

При първото извикване на обвивката на Gradle, правилната версия на Gradle ще бъде изтеглена автоматично. Необходима е интернет връзка, за да работи.

Генерираният APK файл се записва в android/build/outputs/apk като android-debug.apk

# **Заключение**

Botanica Wellness е комплексна платформа за мониторинг и контрол на болестите по растенията в домашни и индустриални условия, която демонстрира силата и напредъка на новите технологии в земеделието, като в същото време успява да го постигне с напълно отворен код, което от своя страна допринася за развитието на подобни технологии и сформирането на общество в тази сфера. Проектът представлява вълнуваща възможност за създаване на иновативно и полезно решение за градинарите и любителите на растенията. Проектът има потенциал да съчетае технологичния напредък с нуждите на домашните градинари, като им предостави инструмент за ранно откриване и управление на болестите по растенията, което ще допринесе за подобряване на здравето и красотата на техните растения.

Въпреки че проектът се сблъсква с предизвикателства в различни аспекти като технологията, дизайна, интеграцията и други, той носи в себе си възможност за създаване на ценно и иновативно устройство, което може да бъде от полза за домашните градинари и общността като цяло.

Проектът показва напредък във възможностите за устойчиво земеделие и PA. Чрез използването на технологии като Интернет на нещата и безжични сензорни мрежи, проектът отваря нови възможности за проследяване и контрол на заболяванията по растенията в реално време.

Отворената сила на софтуера позволява на различни общности и експерти да сътрудничат за развитието и подобряването на приложението, като същевременно гарантира прозрачност и достъпност за всички заинтересовани страни. Това не само насърчава иновациите и обмена на знания, но и ускорява развитието на решенията за борба с болестите по растенията.

# **Информационни източници**

[**https://aws.amazon.com/what-is/iot/#:~:text=The%20term%20IoT%2C%20or%20Internet,as%20between%20the%20devices%20themselves.**](https://aws.amazon.com/what-is/iot/#:~:text=The%20term%20IoT%2C%20or%20Internet,as%20between%20the%20devices%20themselves.)

[**ESP32 Wi-Fi & Bluetooth SoC - Espressif SystemsEspressifhttps://www.espressif.com › products › socs › esp32**](https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.espressif.com/en/products/socs/esp32%23:~:text%3DESP32%2520is%2520highly%252Dintegrated%2520with,Hybrid%2520Wi%252DFi%2520%2526%2520Bluetooth%2520Chip&ved=2ahUKEwjmrOPpqOaFAxWLR_EDHVaqAxIQFnoECBsQAw&usg=AOvVaw1-DxtqIKk7INLRZmmAdLDA)

[**The Internet of Things with ESP32The Internet of Things with**](https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=http://esp32.net/&ved=2ahUKEwjmrOPpqOaFAxWLR_EDHVaqAxIQFnoECBQQAQ&usg=AOvVaw1D6azKz_-EzV7Xdm8Ic7Nw) [**ESP32http://esp32.net**](ESP32http://esp32.net)

[**PowerEdge R720Dellhttps://i.dell.com › sites › data-sheets › Documents**](https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://i.dell.com/sites/content/shared-content/data-sheets/en/Documents/Dell-PowerEdge-R720-Spec-Sheet.pdf&ved=2ahUKEwiFxrr1qOaFAxX-VPEDHb97AroQFnoECAgQAQ&usg=AOvVaw034B7wj75Hl0DkZCUWnrjo)

[**NGINX: Advanced Load Balancer, Web Server, & Reverse**](https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.nginx.com/&ved=2ahUKEwidkuH8qOaFAxWJYPEDHbFWAJAQFnoECAcQAQ&usg=AOvVaw10RW2cXcmCuZ2YnsYWHFKR) [**ProxyNGINXhttps://www.nginx.com**](ProxyNGINXhttps://www.nginx.com)

[**MDN Web DocsMDN Web**](https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://developer.mozilla.org/&ved=2ahUKEwjAwpiFqeaFAxWsVPEDHYQZBdUQFnoECBEQAQ&usg=AOvVaw2SKyx0njvBQRAGWeriS8JP) [**Docshttps://developer.mozilla.org**](Docshttps://developer.mozilla.org)

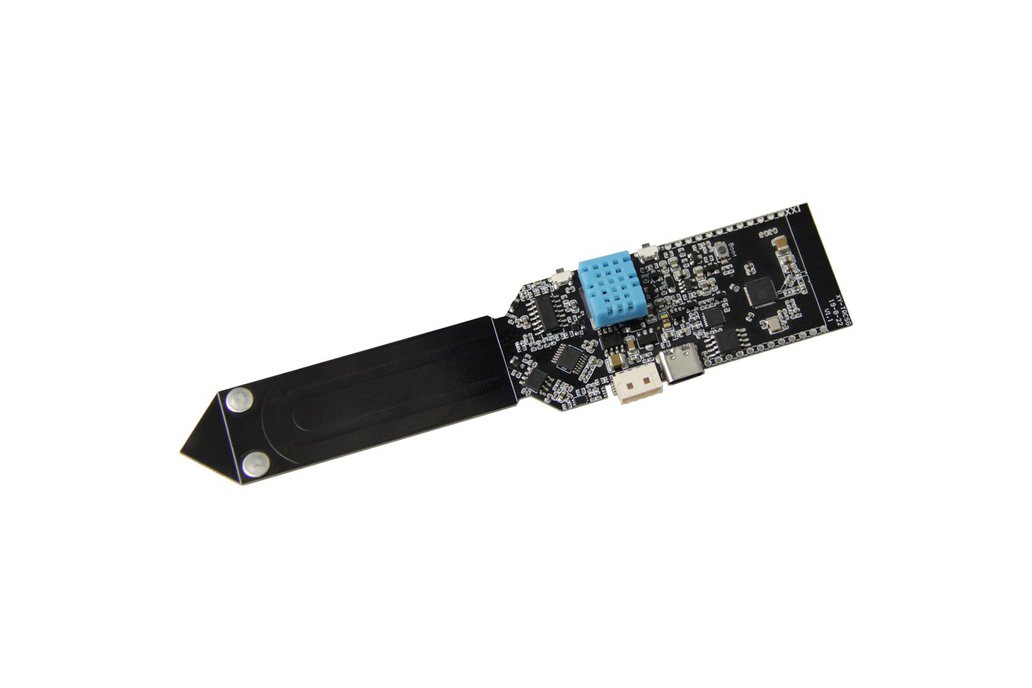
[**Kotlin for Android | Kotlin DocumentationKotlinhttps://kotlinlang.org › ... › Kotlin for Android**](https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://kotlinlang.org/docs/android-overview.html&ved=2ahUKEwjJqe2qqeaFAxXOVfEDHWUFAHwQFnoECBMQAQ&usg=AOvVaw0HTIHA5Wba-yUXpWxb_ETD)

[**Precision Agriculture: Benefits and Challenges for Technology ...Government Accountability Office (GAO) (.gov)https://www.gao.gov › products › gao-24-105962**](https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.gao.gov/products/gao-24-105962%23:~:text%3DPrecision%2520agriculture%2520involves%2520collecting%252C%2520analyzing,as%2520through%2520reduced%2520input%2520costs.&ved=2ahUKEwiIlrvVqeaFAxUNQ_EDHQ5sA9sQFnoECBEQAw&usg=AOvVaw0H1oCoknnNEh3AVxoHgB-E)

# **Приложения**

**A diagram of a computer network

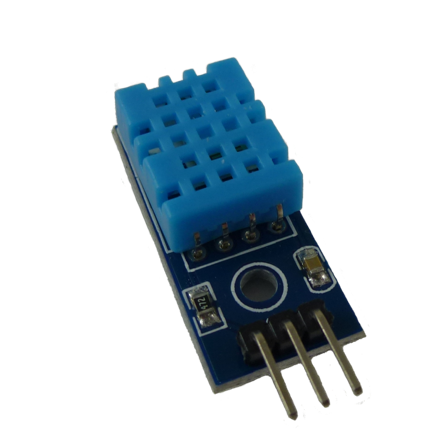
Description automatically generated(Фигура 19 – Схема на работа на Botanica Wellness платформата)**



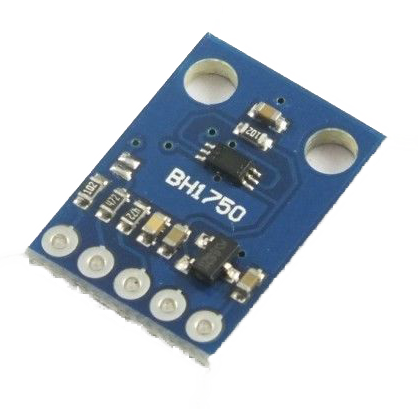
**(Фигура 20 – Основата на IoT устройството ESP32 със свързани сензори)**



**(Фигура 21 – BWS – Dell PowerEdge720 – 2U)**

****

**(Фигура 22 – DHT11 Sensor)**

****

**(Фигура 23 – BH1750 Sensor)**