

УДК 004.384

РАЗРАБОТКА ЭЛЕМЕНТОВ СИСТЕМЫ «УМНЫЙ ДОМ»**Федоров Николай Андреевич¹, Бархатова Ирина Александровна²**¹Студент;

Государственный университет «Дубна»;

141980, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: nirkfed1444@gmail.com.

²Старший преподаватель;

Государственный университет «Дубна»;

141980, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: biriska@mail.ru.

Концепция «Интернета вещей» и мобильная разработка активно развиваются в современной сфере информационных технологий, в частности мобильные устройства под операционной системой Android занимают лидирующее положение по популярности среди других операционных систем для мобильных устройств. Целью данной работы является объединение этих двух направлений разработки в одном проекте для создания информационной системы для обнаружения признаков движения в помещении и оповещения пользователя. В результате была спроектирована система из трех элементов: устройства с датчиком движения, сервера и мобильного приложения.

Ключевые слова: интернет вещей, умный дом, мобильная разработка, встраиваемые системы.

Для цитирования:

Федоров Н.А., Бархатова И.А. Разработка элементов системы «Умный дом» // Системный анализ в науке и образовании: сетевое научное издание. 2021. № 2. С. 32–42. URL : <http://sanse.ru/download/434>.

THE DEVELOPMENT OF THE «SMART HOUSE» SYSTEM'S ELEMENTS**Fedorov Nikolay A.¹, Barkhatova Irina A.²**¹Student;

Dubna State University;

19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;

e-mail: nirkfed1444@gmail.com.

²Senior teacher;

Dubna State University;

19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;

e-mail: biriska@mail.ru.

The “Internet of things” and mobile app development are actively expanding in the current field of IT, in particular mobile devices powered by the Android OS are leading in popularity among other operational systems for smartphones. The goal of this work is to group up these fields under one project the result of which is an information system that detects movement in a room and notifies the user. The end product is a system that consists of 3 elements: a device with a motion sensor, a server and a mobile app.

Keywords: Internet of things, smart house, mobile development, embedded systems.

For citation:

Fedorov N., Barkhatova I. The development of the «Smart house» system's elements. System Analysis in Science and Education, 2021;(2):32–42(In Russ). Available from: <http://sanse.ru/download/434>.

Введение

«Интернет вещей» – концепция, возникшая в 1999 году, и активно развивающаяся по сей день, суть которой заключается во взаимодействии «вещей» (физических и виртуальных устройств) между собой без необходимости непосредственного участия человека в данном процессе [1-2]. Рынок решений, разработанных по данной концепции, активно растет: прогнозируется, что размер рынка интернета вещей достигнет 1.5 трлн долларов США к 2025 году [3].

«Интернет вещей» является одним из приоритетных направлений разработки программного обеспечения (ПО) в мире, которое предоставляет пользователям множество различных решений, в частности разработки т. н. технологии «Умный дом», общий смысл которой можно сформулировать как автоматизация повседневных дел человека. Примерами данной технологии служат системы автоматического включения/выключения света в помещении, зависящие от нахождения человека в комнате, системы автоматического полива и т. д.

Помимо «Интернета вещей» и «Умного дома» другим активно развивающимся направлением является мобильная разработка: в 2020 году число владельцев смартфонов достигло отметки в 3.6 миллиарда человек. Прогнозируется, что к 2023 году эта отметка достигнет 4.3 миллиардов [4]. Таким образом, разработка приложений для мобильных устройств охватывает огромное количество пользователей, делая ее приоритетным направлением разработки. Другим фактором, определяющим направление разработки, является операционная система, под которую будет выполняться разработка. В данном случае операционная система *Android* занимает большую часть рынка мобильных устройств – 71.9% всего рынка по статистике за февраль 2021 [5]. Активная поддержка разработчиков под операционную систему *Android* со стороны компании *Google* делает данное направление наиболее привлекательным для разработчика, желающего начать написание приложений для мобильных устройств.

В качестве языка разработки мобильных приложений для операционных систем *Android* активно набирает популярность язык *Kotlin*, который был выбран главным языком разработки для *Android* на конференции *Google* в 2019 году [6].

Работа, которая будет рассматриваться в данном документе, объединяет вышеописанные направления: система «Умный дом» предназначена для постановки информации об обнаружении признаков движения в помещении. Пользователь будет информироваться с помощью мобильного приложения. Также идентификация «свой-чужой» каждого полученного сигнала будет также осуществляться пользователем в мобильном приложении. Система также является расширяемой в контексте получаемых пользователем данных с устройства посредством добавление новых датчиков.

Данная система состоит из 3 главных элементов: устройство, которое получает показания и отправляет их на сервер, непосредственно сам сервер, получающий данные с устройства и выполняющий их запись в базу данных, и мобильное устройство пользователя, отправляющее запрос на сервер на получение данных и подтверждение идентификации.

Описание формата выходных данных

- описание выходных сообщений

Результатом успешной работы системы будут таблицы полученных сообщений с их соответствующим статусом подтверждения, который будет храниться на сервере, и с изображением, полученным с камеры в момент получения сигнала с датчика движения. Каждое сообщение также имеет свой идентификационный номер, который скрыт от пользователя и используется для последующего подтверждения сигнала пользователем.

- описание структурных единиц информации выходных сообщений

Каждая запись имеет свои время и дату получения сигнала, изображение, и статус подтверждения. Дата и время имеют формат ДД.ММ.ГГГГ и ЧЧ:ММ соответственно, где ДД – номер дня месяца, ММ – номер месяца в году, ГГГГ – год, ЧЧ – час, ММ – минуты.

Изображение имеет формат *JPEG* и размеры 320*240 пикселей.

Описание формата входных данных

- описание входных сообщений

Идентификационный номер записи присваивается сообщению по мере прибытия на сервер, используется для смены статуса записи при подтверждении ее пользователем.

- описание структурных единиц информации входных сообщений

Каждая запись полученных показаний имеет свой идентификационный номер, который отправляется приложением при нажатии пользователем кнопки «Подтвердить», что принимается сервером как подтверждение сигнала. Идентификационные номера задаются четырехзначными числами.

Обзор существующих решений

Из-за природы проекта существующие решения будут включать себя разработки, как крупных компаний, так и простых пользователей-любителей, из-за общей доступности и относительной дешевизны базовых комплектующих подобного рода проектов «Интернета вещей».

ESP8266 PIR Home Security & Notifier

Проект пользователя *rahuladitya303*¹ представляет собой систему оповещения пользователя об обнаружении признаков движения с помощью датчика движения *HC-SR501 PIR* и платы *NodeMCU* со встроенным модулем *Wi-Fi*. Для получения уведомлений на почту пользователю необходимо зарегистрироваться в стороннем приложении *Blynk App*, после чего выдается уникальный токен, который указывается в коде для платы при отправке данных на сервер приложения.

Список компонентов проекта:

- Плата *NodeMCU* – средняя цена – от 155 рублей (на 05.05.2021)²;
- Датчик движения *HC-SR501 PIR* – средняя цена – 75 рублей (на 05.05.2021)³;
- Дополнительные кабели для подключения компонентов;

Преимущества:

- Простота и низкая цена проекта (~230 рублей + затраты на кабели и/или плату разработки): все составляющие имеют низкую цену, код для платы находится в открытом доступе;
- Приложение работает с различными типами плат;
- Небольшие габариты устройства (49*24.5*13 мм);
- Недостатки:
- Отсутствие камеры;
- Отсутствие истории;
- Требуется использование стороннего приложения;
- Оповещение происходит через электронную почту, а не через само приложение;
- Проект не предусматривает независимый источник питания (плата питается за счет подключенного к компьютеру кабеля *micro USB*);

Extaum WiFi Burglar Alarm

Данное решение компании *Extaum*⁴ представляет собой охранное устройство, имеющее датчик движения *PIR* и использующее 3 *AAA* батарейки в качестве источника питания.

¹ <https://www.hackster.io/rahuladitya303/esp8266-pir-home-security-notifier-b8245c>

² <https://aliexpress.ru/item/32665100123.html>

³ <https://aliexpress.ru/item/32699216549.html>

⁴ <https://www.amazon.com/Extaum-Infrared-Detector-Compatible-Security/dp/B07TSGWGP7>

Для оповещения пользователя об обнаружении сигнала используются стороннее приложение – *Smart Life/Tuya* (доступно для ОС *iOS* и *Android*).

Устройство имеет 2 режима работы: стандартный и экономный, с получением показаний каждые 2 минуты или каждые 4 минуты соответственно.

Стоит отметить следующие характеристики:

- Габариты: 6.5*6.5*2.9 см;
- Вес: 51 г;
- Цена: от 19.79 Долларов США (~1479.15 на 05.05.2021);

Преимущества:

- Готовый коммерческий продукт – от пользователя требуется только установка устройства и регистрация в приложении;
- Приложение поддерживает разные операционные системы;
- Получение уведомлений с приложения;
- Возможность просмотра истории полученных уведомлений;
- Разные настройки работы;
- Поддержка работы с *Alexa* и *Google Home*;

Недостатки:

- Относительно высокая стоимость по сравнению с рассмотренными выше проектами (~1479 рублей против ~225);
- Отсутствие камеры;
- Использование стороннего приложения;

Выводы

Таким образом, после рассмотрения существующих решений были выделены следующие требования, которые должны будут учтены при разработке системы:

- Наличие камеры – устройство должно иметь камеру для получения изображений в качестве дополнительной проверки от ложных сигналов;
- Небольшие габариты – общие размеры устройства не должны превышать 70*70*30 мм;
- Относительно низкая цена – цена всех компонентов не должна превышать 1000 рублей для обеспечения доступности данного решения;
- Для получения данных и оповещения пользователя должно быть создано отдельное приложение, без использования сторонних сервисов;
- Возможность просмотра пользователем всей истории полученных уведомлений.

Проектирование архитектуры

Для упрощения описания архитектуры элементы системы были представлены в виде классов, имеющих свои методы и поля:

- *Device* – класс устройства, содержит в себе поля для идентификационного номера устройства, изображения и статуса обнаружения движения, имеет метод отправки сообщений на сервер.
- *Server* – класс сервера, содержит в себе все полученные сообщения с устройств со статусом подтверждения их пользователем, а также, к какому пользователю они принадлежат. Содержит методы для получения сообщений, отправки истории пользователю и обновление статуса записи.
- *App* – класс приложения, одержит идентификационный номер пользователя и методы для получения истории и подтверждения записи.

Устройство

Концепция «интернета вещей» тесно связана с понятием «встраиваемая система» (*Embedded systems*) – специализированная микропроцессорная система, программный код которой разработан на

выполнение конкретной функции как независимая система или как подсистема [7]. В данной работе встраиваемой системой будет являться устройство, получающее показания с датчика движения и передающее данные на сервер.

В качестве устройства был выбран одноплатный модуль *ESP32-CAM*, специально разработанный для разработки устройств «интернета вещей» и работающий на базе *ESP8266* – микроконтроллера с интерфейсом *Wi-Fi*.

Отличительной особенностью *ESP32-CAM* (Рис. 1), из-за которой он был выбран для использования в данной работе, является специально выделенный разъем для камеры. Устройство также имеет разъем для карт типа *MicroSD*, однако, добавление карты памяти также блокирует доступ к цифровым выходам платы, так как они выделяются для работы с картой.

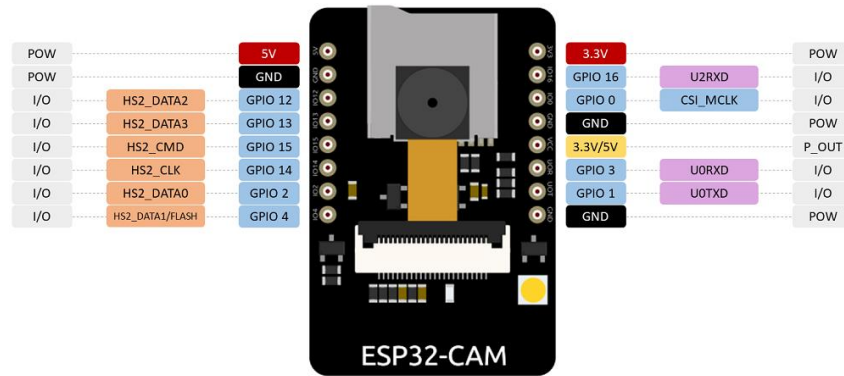


Рис. 1. *ESP32-CAM*

Характеристики [8][9]:

- Вес: 5 г;
- Габариты: 27*40.5*4.5 мм;
- Напряжение питания: 5 В;
- Ток потребления: 180 мА;
- Тактовая частота процессора: 2 ядра*160 МГц;
- Память: 520 КБ *SRAM*, 4МБ *PSRAM*;
- Беспроводная связь: *Bluetooth 4.2 BR/EDR*, *Wi-Fi 802.11 b/g/n 2.ГГц*;

В комплекте с *ESP32-CAM* часто идет камера *OV2640* (Рис. 2), которая использовалась в системе.



Рис. 2. Камера *OV2640*

Характеристики [10][11]:

- 2-мегапиксельная камера;
- автоматический контроль экспозиции, автоматический контроль усиления, автоматический баланс белого, автоматическое устранение световых полос, автоматическая калибровка черного уровня. Контроль качества изображения, включая насыщенность цвета, оттенков, гамма, резкость;
- Разрешение матрицы датчика изображения: 1600*1200 (*UXGA*);
- Питание: 3.3 В;

- Уровни напряжений *IO*: 1.7 В ~ 3.3 В (постоянный ток);
- Угол обзора: 70 градусов;

В качестве датчика движения в данном проекте использовался пирозлектрический датчик *HC-SR501 PIR* – малогабаритный, простой в использовании и недорогой вариант датчика движения, что является причиной его выбора.

Характеристики датчика [12][13][14]:

- Работает при разных напряжениях: от 4 до 12В (рекомендуемое напряжение – 5В и выше);
- Различает движения объектов и движения человека;
- Имеет два режима работы: *Repeatable(H)* – получение показаний через настраиваемый промежуток времени – и *Non-Repeatable(L)* – при обнаружении движений человека датчик будет подавать сигнал пока человек не выйдет из его поля зрения;
- Угол обзора – 120 градусов, максимальное расстояние – 7 метров;
- Низкое потребление тока – 65 мА;

Датчик имеет 3 пина: питание, земля и цифровой выход. Подключение к плате *ESP32-CAM* осуществляется следующим образом (Рис. 3).

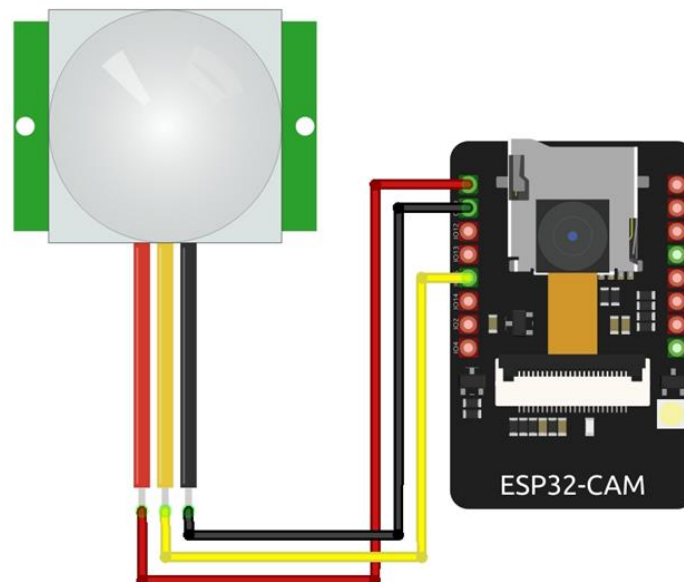


Рис. 3. Схема подключения

Программирование. Программатор *FTDI FT232RL*

Так как *ESP32-CAM* не имеет разъема для подключения к компьютеру, для программирования платы требуется программатор.

Конвертер *USB-UART* (Рис. 4) на основе микросхемы *FT232RL* предназначен для организации виртуального *COM*-порта с уровнями *TTL* для взаимодействия с внешними устройствами [15].

Характеристики [15][16]

- Вес: 5 г;
- Габариты: 43 × 18 × 12 мм;
- Рабочее напряжение: 3,3 и 5 В;
- Потребляемый ток: 50 мА;
- Интерфейс: *mini USB*;

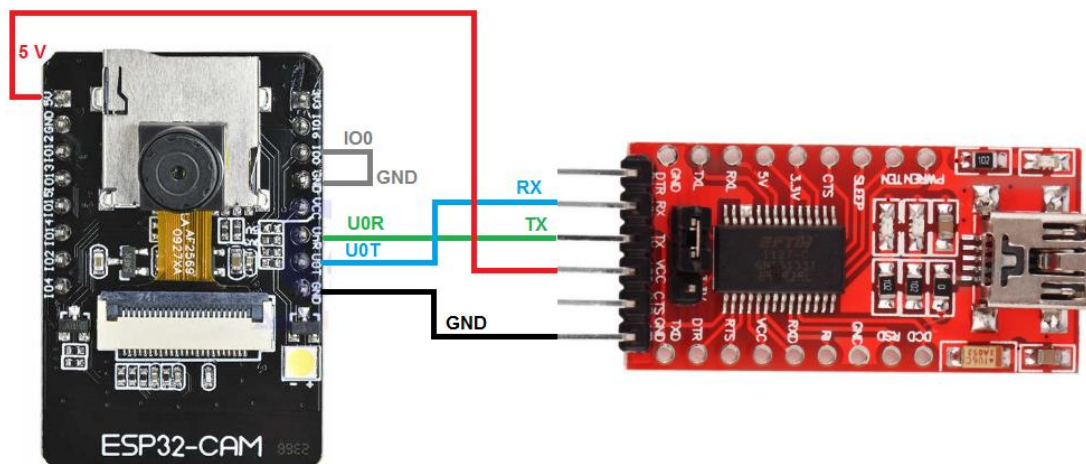


Рис. 4. Схема подключения

Серверная часть

База данных

База данных приложения состоит из двух таблиц (Рис. 5). Первая хранит список пользователей с закрепленными за ними устройствами. Это было сделано за тем, чтобы в будущем система была расширяемой и поддерживала работу со многими пользователями. Вторая таблица хранит историю всех событий: от какого устройства поступил сигнал, время поступления сигнала и был ли сигнал подтвержден пользователем (данное поле имеет логический тип данных и имеет значение по умолчанию *False*).

EventHistory		Users	
id	int	id	int
device_id	int	user_id	int
date	varchar(20)	device_id	int
time	varchar(20)	device_name	varchar(50)
image	varchar(100)		
isConfirmed	boolean		

Рис. 5. База данных

В качестве СУБД в данном проекте использовалась *PostgreSQL* – свободная объектно-реляционная СУБД, находящаяся в открытом доступе.

Node.js, Express

Node.js представляет среду выполнения кода на *JavaScript*, которая построена на основе движка *JavaScript Chrome V8*, который позволяет транслировать вызовы на языке *JavaScript* в машинный код. *Node.js* прежде всего предназначен для создания серверных приложений на языке *JavaScript*.

В свою очередь, *Express* является фреймворком веб-приложений для *Node.js* и в данном проекте будет использоваться для создания *API* для сервера. Причиной выбора данной технологии для создания *API* обуславливается относительной простотой разработки.

Разработка приложения

При разработке приложения использовался шаблон проектирования *Model-View-ViewModel* (*MVVM*), суть которого заключается в разделении объектов на 3 группы [17]:

- *Model* – в данной группе хранятся вся бизнес-логика и данные;
- *View* – отвечает за презентацию данных и средств управления на экране;
- *View model* – преобразует данные из модели в значения, которые может отобразить *View*;

В приложении этот паттерн осуществляется через использование классов *Fragment* и *ViewModel*. Класс *Fragment* отвечает за презентационный слой, *ViewModel* – по назначению идентичен одноименной группе объектов из паттерна *MVVM*.

Для отправки запросов на сервер была использована библиотека *Retrofit* – типобезопасный *HTTP* клиент, который значительно упрощает получение данных благодаря преобразованию *HTTP API* в интерфейс *Java/Kotlin* и обработке данных, полученных от сервера в формате *JSON* [18].

Для загрузки изображений с сервера была использована библиотека *Glide*, использующаяся для асинхронного получения изображений из сети, ресурсов, файловой системы, их кэширования и отображения [19].

Интерфейс приложения представлен на Рис. 6. Главный экран, который видит пользователь после открытия приложения, отображает полученные устройством сигналы за последний день с соответствующими кнопками для их подтверждения и с полученным изображением с камеры в момент получения сигнала. При подтверждении конкретного сигнала кнопка заменяется соответствующим символом, отображающим, что подтверждение получено. «История событий» содержит в себе всю историю полученных сигналов с их статусом подтверждения, отсортированную по времени и дате. «Настройки» позволяют отключить получение сигналов с устройства, включить уведомления на смартфоне и содержат в себе опцию очистки истории. Пользователь также имеет возможность сортировать записи по устройству, с которого был получен сигнал.

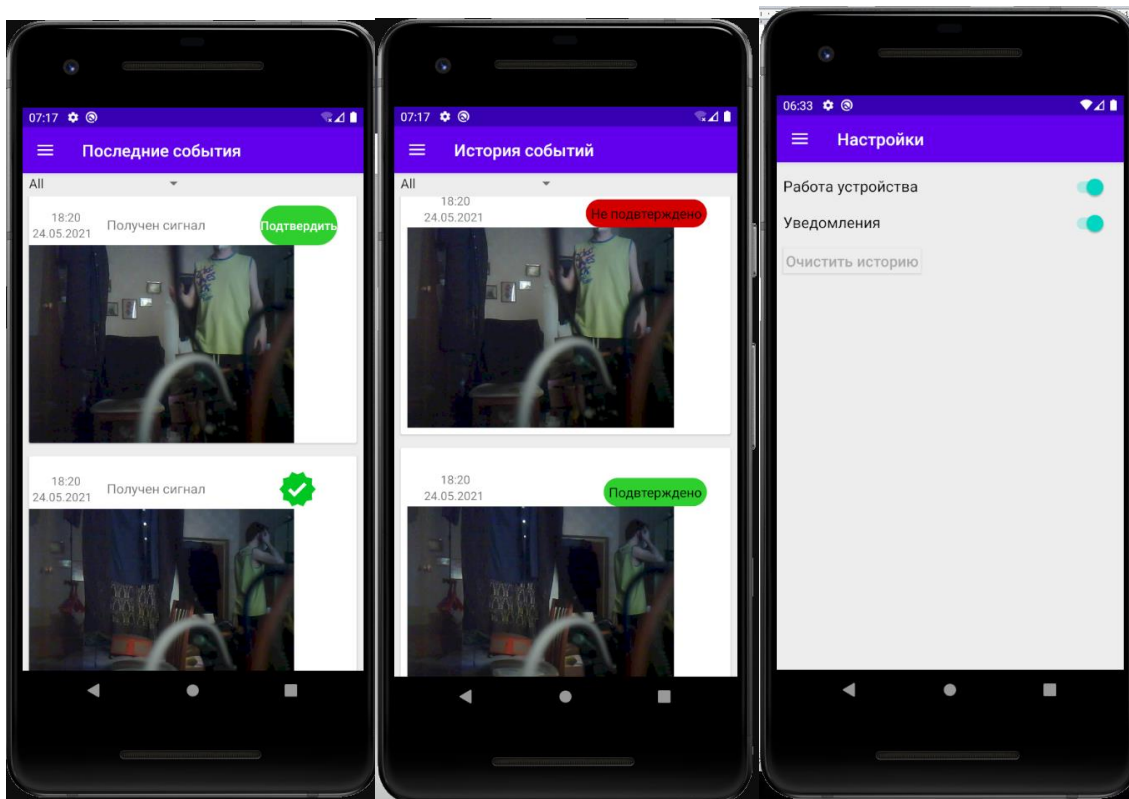


Рис. 6. Экраны приложения

Рассмотрение результатов

В результате разработки была создана система, полностью соответствующая изначальным требованиям:

- Размеры устройства не превышают поставленные 70*70*30 мм (*HC-SR501 PIR*: 32.2*24.3*25.4 мм, *ESP32-CAM*: 27*40.5*4.5 мм);
- Цена не превышает 1000 рублей (*ESP32-CAM*: ~500 рублей⁵, *HC-SR501 PIR*: ~75 рублей⁶, *FTDI FT232RL* – 289 рублей [15]);
- Устройство снабжено камерой;
- Для просмотра данных было разработано отдельное приложение.

Разработанная система успешно справляется с поставленной задачей, позволяя пользователю выполнять установленные в требованиях действия такие, как просмотр последних сообщений устройства с изображениями с камеры, полученными в момент получения сигнала с датчика движения, просмотр истории записей, подтверждение сообщений, сортировка записей по устройству, настройка уведомлений, включение/выключение получения новых сообщений с сервера, очистка истории сообщений.

Для работы системы требуется подключение к сети Интернет всех ее элементов. Также обязательна регистрация пользователя и его устройств перед использованием системы.

Перспективы развития

Несмотря на соответствие поставленным требованиям, данная система имеет моменты, которые могут быть улучшены: например, добавление новых элементов интерфейса приложения, позволяющие отсортировать полученный список по статусу («Подтвержденные»/ «Не подтвержденные»), добавление настройки типа уведомлений («Компактные»/ «Подробные» (с уменьшенным изображением)) и т. д.

Также стоит отметить, что на данный момент в системе отсутствует средство регистрации помимо прямого редактирования базы данных на сервере. Имплементация подобного элемента обязательна в случае применения системы не только для личного пользования. Регистрацию пользователя можно реализовать в виде отдельной формы на сервере, либо при первом входе в приложение. Одним из вариантов регистрации устройства является получение идентификационного номера, генерируемого на сервере, что применялось в рассмотренном решении *ESP8266 PIR Home Security & Notifier*. Данный номер, так же, как и в проекте, далее следует указать непосредственно в коде самого устройства.

Так как серверной части не принципиален тип устройства, с которого получают сообщения, одним из вариантов развития работы является поддержка других типов плат, таких как *Arduino*, *NodeMCU* и *Raspberry Pi*, которые рассматривались в качестве используемого в проекте типа платы. Однако следует взять во внимание, что не все типы камер поддерживаются каждой из плат: *Raspberry Pi* имеет свой тип камеры, который не поддерживается *Arduino* и *NodeMCU*.

Весь исходный код проекта будет загружен в открытый репозиторий на *GitHub*⁷, что позволит другим пользователям использовать и модифицировать код данной работы.

⁵ <https://aliexpress.ru/item/1005001978372728.html>

⁶ <https://aliexpress.ru/item/32699216549.html>

⁷ <https://github.com/Kola14/MyHomeApp>, <https://github.com/Kola14/MyHomeAppExtra>

Заключение

Таким образом, созданная система справляется со своей задачей в предоставлении пользователю сообщений об обнаружении признаков движения в помещении с отображением соответствующего изображения, полученного устройством в момент получения сигнала с датчика движения. Устройство состоит из недорогих и общедоступных компонентов, которые может приобрести любой желающий для имитации данного проекта. От пользователя лишь требуется создание собственного сервера с использованием сервисов хостинга (как это было сделано в данном проекте) или использования отдельной уже имеющейся машины в качестве сервера.

Разработанная система имеет множество потенциалов развития такие, как доработка возможностей графического интерфейса приложения, поддержка других типов устройств таких, как *Raspberry Pi* и другие и т.д. Любой пользователь может просмотреть исходный код приложения и использовать его и/или внести правки в программный код.

Список литературы

1. Alexander S. Gillis. internet of things (IoT), 2020. URL: <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>.
2. International Telecommunication Union. ITU-T Y.4000/Y.2060, 2012. URL: <http://handle.itu.int/11.1002/1000/11559>.
3. Lionel Sujay Vailshery. Forecast end-user spending on IoT solutions worldwide from 2017 to 2025, 2019. URL: <https://www.statista.com/statistics/976313/global-iot-market-size/#:~:text=The%20global%20market%20for%20Internet,around%201.6%20trillion%20by%202025>.
4. S. O'Dea. Number of smartphone users worldwide from 2016 to 2023, 2020. URL: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>.
5. StatCounter. Mobile Operating System Market Share Worldwide, 2021. URL: <https://gs.statcounter.com/os-market-share/mobile/worldwide#>.
6. Android Developers. Android's Kotlin-first approach, 2021. URL: <https://developer.android.com/kotlin/first>.
7. Systems, 2021. URL: <https://www.omnisci.com/technical-glossary/embedded-systems>.
8. Электронный магазин «Вольтик». ESP32-CAM – плата разработки ESP32 с камерой OV2640, 2021. URL: <https://voltiq.ru/shop/esp32-cam-wi-fi-module/>.
9. Shenzhen Ai-Thinker Technology Co., Ltd. ESP32-CAM Module, 2017. URL: <https://loboris.eu/ESP32/ESP32-CAM%20Product%20Specification.pdf>.
10. Электронный магазин «Robotclass». Камера OV2640, 2021. URL: https://shop.robotclass.ru/index.php?route=product/product&product_id=1559.
11. OmniVision Technologies. OV2640 Color CMOS UXGA (2.0 MegaPixel) CameraChip with OmniPixel2 Technology, 2006. URL: https://www.waveshare.com/w/upload/6/6b/OV2640_DS%281.6%29.pdf.
12. Adafruit Industries. PIR Motion Sensor, 2014. URL: <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor>.
13. Components101. HC-SR501 PIR Sensor, 2017. URL: <https://components101.com/sensors/hc-sr501-pir-sensor>.
14. Adafruit Industries. PIR Motion Sensor, 2020. URL: <https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>.
15. Электронный магазин «Вольтик». FTDI FT232RL – USB – UART TTL преобразователь, 2021. URL: <https://voltiq.ru/shop/usb-uart-converter-ftdi-ft232rl/>.

16. Future Technology Devices International Ltd. FT232R USB UART IC Datasheet, 2020. URL: https://ftdichip.com/wp-content/uploads/2020/08/DS_FT232R.pdf.
17. Jay Strawn. Design Patterns by Tutorials: MVVM, 2018. URL: <https://www.raywenderlich.com/34-design-patterns-by-tutorials-mvvm>.
18. Square, Inc. Retrofit - A type-safe HTTP client for Android and Java, 2021. URL: <https://square.github.io/retrofit/>.
19. Sam Judd. Glide, 2021. URL: <https://github.com/bumptech/glide>.