

ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СЕРВИСА КОНТРОЛЯ АКТИВНОСТИ В ПОМЕЩЕНИИ

Карпушина С.К., Копылова А.В., Миронов А.Н.

*МИРЭА - Российский технологический университет, 119454, Россия, г. Москва, проспект Вернадского, 78,
e-mail: sofyakarp1@yandex.ru, aspyver@gmail.com, amironov1993@yandex.ru*

Рассмотрено понятие «Интернет вещей», определены задачи сервиса, спроектирована архитектура сервиса, описан каждый компонент сервиса. Приведен анализ аппаратной части сервиса, на основании анализа выбрана аппаратная часть сервиса.

Ключевые слова: интернет вещей, мобильное приложение, облачная платформа, база данных.

DESIGNING A SOFTWARE SERVICE FOR MONITORING INDOOR ACTIVITY

Karpushina S. K., Kopylova A.V., Mironov A.N.

*MIREA - Russian Technological University, 78 Vernadsky Avenue, Moscow 119454, Russia
e-mail: sofyakarp1@yandex.ru, aspyver@gmail.com, amironov1993@yandex.ru*

The concept of "Internet of Things" is considered, the tasks of the service are defined, the architecture of the service is designed, and each component of the service is described. The analysis of the hardware part of the service is given, based on the analysis, the hardware part of the service is selected.

Keywords: internet of things, mobile app, cloud platform, database.

Введение

Бурное развитие информационных технологий способствует автоматизации практически всех сфер человеческой деятельности. Внедрение сервиса мониторинга активности человека увеличит безопасность нахождения человека с ограниченными возможностями в помещении. С помощью различных способов сбора данных об активности, обработки данных, быстрого и удобного отображения результата обработки, можно получать актуальную информацию об наблюдаемом человеке. Различные технологии позволяют создать единый сервис, состоящий из базы данных, пользовательского мобильного приложения, серверной части, подключения к облачной платформе и аппаратной части.

Для реализации данного сервиса необходимо:

1. Уточнить цели и задачи к разрабатываемому программному сервису
2. Описание компонентов сервиса
3. Выбрать аппаратную часть
4. Спроектировать архитектуру сервиса

Цели

Программный сервис контроля активности в помещении для помощи людям с ограниченными возможностями предназначен для повышения безопасности при нахождении людей (люди с ограниченными возможностями, пенсионеры, дети) в помещении без посторонней помощи. Данный сервис оповещает пользователя об активности в помещении или наоборот ее отсутствии, собирает и обрабатывает информацию об активности, а также хранит информацию за определенный период.

Цель работы - повышение безопасности людей в помещении за счет учета действий пользователя и оповещении об опасном состоянии. Опасным считает состояние – когда пользователь находится в помещении, но при этом долгое время не совершает никаких действий.

Задачи

Мобильное приложение должно иметь:

- возможность настройки оповещений;
- возможность настройки разных режимов;
- возможность отключения наблюдаемых пользователей;
- возможность изменения персональных данных;

Задачи сбора и хранения данных

Вся информация, поступающая от облачной платформы к мобильному приложению, должна храниться в базе данных до срока истечения периода хранения или до удаления пользователя. При изменении персональных данных пользователя, новая и неактуальная информация хранятся в базе данных. Это необходимо для оперативного доступа к информации за прошедший период. Так же необходимо разработать сценарий, при котором происходит сбой при передаче информации от базы данных к мобильному приложению, информация не должна удаляться.

Задачи анализа данных

Вся информация, поступающая от облачной платформы к мобильному приложению, должна анализироваться с помощью разработанных алгоритмов на стороне приложения. Датчики и облачная платформа не поддерживают анализ или хранение информации.

Определение потоков данных системы



Рис. 1 Определение архитектуры сервиса

На диаграмме выше представлено 4 компонента сервиса:

1 уровень

- Датчики собирают данные и отправляют ее на облачную платформу.

2 уровень

- Облачная платформа – связующее звено между датчиками и пользователем. Необходимый элемент для доступа к датчикам и приложению. Она получает данные, сохраняет их и ждет запроса от сервера, чтобы передать уже информацию об активности.

3 уровень

- Сервер делает запрос на облачную платформу, а в ответ получает информацию об активности от датчика. Обрабатывает ее с помощью разработанных алгоритмов и в ответ на запрос с пользовательского мобильного приложения выдает результат, активен или неактивен, а также, в случае опасной ситуации, номера экстренных служб.

4 уровень

- Пользовательское мобильное приложение – компонент для отображения актуальной информации, предупреждения пользователя об опасной ситуации или отображение информации за прошедший период. Приложение делает запрос на сервер и получает ответ, отображая его.
- База данных – компонент для хранения информации пользователей. Она хранит в себе все результаты обработки, а также персональную информацию наблюдаемых и наблюдающих пользователей.

Общее описание компонентов системы

Микроконтроллер

Микроконтроллер является микросхемой. Он основан на аналоговом принципе действия. Система понимает лишь две команды («есть сигнал», «нет сигнала»). Из этих сигналов в его память вписывается код определенной команды. Когда МК считывает команду, он ее выполняет.

В каждом из МК прописаны свои базовые наборы команд. И только их он способен принимать и выполнять. Сочетая отдельные команды между собой, можно написать уникальную программу, по которой будет работать любое электронное устройство именно так, как требуется.

Облачные IoT платформы

Облачная платформа IoT — это место, где возможности стеков технологий IoT и облачных вычислений объединяются, чтобы принести дополнительную ценность как для потребительских, так и для бизнес-приложений.

Одним из самых больших преимуществ размещения вашей системы Интернета вещей в облаке является то, что она очень легко масштабируется. В случае сложных локальных сетевых инфраструктур масштабирование требует приобретения большего количества оборудования, инвестиций большего количества времени и увеличения усилий по настройке, чтобы заставить его работать должным образом.

Сервер мобильного приложения

Серверы приложений – это серверы, которые обеспечивают среду со всеми необходимыми требованиями для запуска или разработки приложения. Поскольку серверы обычно используются для предоставления услуг, которые постоянно требуются, большинство серверов никогда не выключаются. Следовательно, когда серверы выходят из строя, они могут вызвать у пользователей сети и компании множество проблем. Чтобы устранить эти проблемы, серверы обычно настраиваются на отказоустойчивость.

Пользовательское мобильное приложение

Мобильное приложение, чаще всего называемое приложением, — это тип прикладного программного обеспечения, предназначенного для запуска на мобильном устройстве, таком как смартфон или планшетный компьютер. Мобильные приложения часто служат для предоставления пользователям услуг, аналогичных тем, которые доступны на ПК.

База данных

База данных — это организованный набор структурированной информации или данных, обычно хранящихся в электронной форме в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД). Вместе данные и СУБД, а также связанные с ними приложения называются системой баз данных, часто сокращенной до просто базы данных.

Вместе компоненты сервиса представляют систему Интернет вещей.

Простыми словами, интернет вещей (IoT) – это объединение устройств в единую компьютерную сеть для сбора, анализа, обработки и передачи данных другим объектам посредством программного обеспечения, приложений и разных технических устройств.

Устройства интернета вещей работают сами по себе, но человек имеет возможность настраивать их и предоставлять доступ к конкретным данным. Они функционируют в онлайн-режиме и состоят из платформы-облака и «умного» оборудования. Подключение к облаку производится посредством Wi-Fi, Bluetooth или иных

форм связи.

Выбор аппаратной части сервиса

Использование аппаратной части в данном сервисе неоспоримо, но необходимо определить, что именно использовать. Существует множество решений по считыванию информации. Давай рассмотрим самые популярные из них.

Первый способ: вмонтировать реле в выключатель света.

Первый способ основан на нажатие на кнопку на выключателе света. Алгоритм данного способа: в выбранные промежутки времени микроконтроллер отправляет на облако информации о том, включен или выключен свет, если свет был включен (реле изменила свое состояние), значит наблюдаемый пользователь подает признаки активности. При каждом изменении состояния реле посылается на облако информация.

Плюсы:

- Точное получение информации (если реле меняет свое состояние, обновленная информация приходит на облако)
- Простота реализации прошивки
- Простота обработки полученной информации

Минусы:

- Необходимость вмонтирования реле в выключатель (в некоторых случаях невозможность вмонтирования реле в выключатель)
- Ложные срабатывания. Допустим вы устанавливаете реле на выключателе на кухню, наблюдаемый человек изменил состояние реле при включении света. Далее продолжительное время состояние не меняет, это означает три сценария развития: либо наблюдаемый пользователь просто не выключил свет при выходе из кухни, либо он еще не выходил из кухни, либо ему стало плохо, и он не в состоянии переключить свет.

Данный способ не гарантирует даже частичную безопасность, так как может возникнуть слишком большой процент ложных срабатываний.

Второй способ: камера слежения.

В комнате наблюдаемого устанавливается камера слежения, передающая все полученные изображения на пользовательское приложение. Это самый точный способ получения информации об активности без ложных срабатываний.

Но данный способ нарушает личные границы человека, его конфиденциальность и безопасность в цифровом пространстве.

Плюсы:

- Маленький процент ложных срабатываний
- Визуальное изображение

Минусы:

- Высокая цена на аппаратуру
- Нарушение конфиденциальности
- Сложность обработки полученной информации
- Снижение безопасности человека в цифровом пространстве

Так же для того, чтобы оперативно и быстро получать сообщения об опасных ситуациях, наблюдающий пользователь должен ежеминутно проверять изображение в мобильном приложении. Этого можно избежать, если полученные видеоданные обрабатывать и выдавать результат обработки пользователю, как в случае с реле, но для этого необходимо применить технологии искусственного интеллекта, нейросети, которые значительно увеличивают процент ложных срабатываний, стоимость и сложность реализации. Данный способ не подходит в

качестве аппаратной части, так как не удовлетворяет требованиям.

Третий способ: датчики движения.

Датчики движения устанавливаются в активно посещаемых местах помещения, и отправляют информацию если движение было, а также каждые 5 минут.

Плюсы:

- Простота реализации
- Точное получение информации
- Простота обработки полученной информации

Минусы

- Большой процент ложных срабатываний (необходимо учитывать много ограничений)

В ходе проведенного анализа было принято решение совместить два способа сбора информации: первый и третий, и на основании их реализовывать программный сервис. Это снижает процент ложных срабатываний.

Технологии компонентов

Информация при передаче должна быть представлена в формате JSON.

Для передачи информации от датчиков к облачной платформе необходимо использовать протокол MQTT.

Необходимо использовать облачную платформу Rightech IoT.

Прошивку для контроллера необходимо написать на языке программирования C++ с использованием специализированных библиотек (выявляются на этапе проектирования) для датчиков.

Датчики для сбора информации необходимо использовать датчики движения (более конкретно выявляется на этапе проектирования).

Сервер мобильного приложения должен быть написан на языке программирования Java с использованием Фреймворков Spring, Vertx, Jersya.

Пользовательское мобильное приложение должно быть написано на языках Java и XML.

Для передачи информации от облачной платформы к серверу мобильного приложения необходимо использовать протокол HTTPS.

Для передачи информации от сервера приложения к пользовательскому мобильному приложения необходимо использовать протокол HTTPS.

В качестве базы данных необходимо использовать Postgre sql.

Мобильное приложение должно быть разработано с помощью среды разработки Android Studio.

Заключение

В ходе проектирование данного сервиса были определены используемые технологии, порядок передачи данных, форматы представления данных. На основании проделанной работы можно приступить к реализации всех компонентов, тестированию, отладке и эксплуатации программного сервиса.

Список литературы

1. Облачные платформы. [Электронный ресурс] — Режим доступа: <https://iot.ru/wiki/oblachnyye-platformy> (дата обращения 10.03.2021).
2. Обзор облачных платформ IoT Обзор облачных платформ IoT. [Электронный ресурс] — Режим доступа: <https://www.edsson.com/ru/blog/article?id=iot-platforms> (дата обращения 08.03.2021).
3. Филипп Б., Стюарт К., Марсикано К., Android. Программирование для профессионалов. 2017, № 2. С 32-39 (дата обращения 09.03.2021).
4. Hello, Android: Introducing Google's Mobile, Development Platform – 2018 г., № 1. С 48-54 (дата

обращения 02.03.2021).

5. Android разработка - с нуля до профессионала | [Электронный ресурс - онлайн курс] — Режим доступа: <https://www.udemy.com/course/android-kak-po-notam-a/learn> (дата обращения 12.03.2021).

6. Rightech IoT Cloud. [Электронный ресурс] — Режим доступа: <https://rightech.io/en/main/> (дата обращения 11.03.2021).

7. Документация Postgres Sql | [Электронный ресурс] — Режим доступа: <https://postgrespro.ru/about/license> (дата обращения 13.03.2021).

References

1. Cloud platforms. [Electronic resource] - Access mode: <https://iot.ru/wiki/oblachnye-platformy> (accessed 10.03.2021).

2. Overview of IoT Cloud Platforms Overview of IoT cloud platforms. [Electronic resource] - Access mode: <https://www.edsson.com/ru/blog/article?id=iot-platforms> (accessed 08.03.2021).

3. Phillips B., Stewart K., Marsicano K., Android. Programming for professionals. 2017, No. 2. From 32-39 (accessed 09.03.2021).

4. Hello, Android: Introducing Google's Mobile, Development Platform-2018, No. 1. From 48-54 (accessed 02.03.2021).

5. Android development - from scratch to professional | [Electronic resource-online course] - Access mode: <https://www.udemy.com/course/android-kak-po-notam-a/learn> (accessed 12.03.2021).

6. Rightech IoT Cloud. [Electronic resource] - Access mode: <https://rightech.io/en/main/> (accessed 11.03.2021).

7. 7. Postgres Sql Documentation | [Electronic resource] - Access mode: <https://postgrespro.ru/about/license> (accessed 13.03.2021).