ВВЕДЕНИЕ

Данная дипломная работа посвящена проектированию web-сервиса для удаленного доступа к программно-аппаратной платформе домашней автоматизации.

Внедрение практических решений для концепции «интернета вещей» начиная с 2010-х годов считается одной из устойчивых тенденций в сфере ИТ. Это напрямую связанно с появлением облачных вычислений, повсеместным распространением беспроводных сетей, освоением программно-конфигурируемых сетей и развитием технологий межмашинного взаимодействия (Machine-to-Machine).

IoT (Internet of Things) - это новая парадигма Internet. Под термином "вещь" в IoT подразумеваются интеллектуальные, т.е. "умные" предметы или объекты. IoT - это традиционная или существующая сеть Интернет, расширенная подключенными к ней вычислительными сетями физических предметов или вещей, которые могут самостоятельно организовывать различные модели подключения или общения ("Thing - Thing", "Thing - User" и "Thing - Web Object").

Следует отметить, что «умные» вещи– это датчики или приводы, снабженные микроконтроллером с ОС реального времени со стеком протоколов, памятью и устройством связи, встроенные в различные объекты, например, в электросчетчики или газовые счетчики, датчики давления, вибрации или температуры, выключатели и т.д. "Умные" объекты могут быть организованны в вычислительную сеть физических объектов, которые могут быть подключены через шлюзы (хабы или специализированные IoT платформы) к традиционной сети Интернет.

На технологическом уровне IoT – это концепция развития инфраструктуры сети (физической основы) Интернет, в которой "умные" вещи без участия человека способны подключиться к сети для удаленного взаимодействия с другими устройствами (Thing - Thing) или взаимодействия с автономными или облачными ЦОДами или DATA-центрами (Thing - Web Objects) для передачи данных на хранение, их обработку, аналитику и принятия управленческих решений, направленных на изменение окружающей среды, или для взаимодействия с пользовательскими терминалами (Thing - User) для контроля и управления этими устройствами.

Концепция облачных вычислений, появилась в 2006 году, когда Amazon.com, в то время книжный интернет-магазин, представил Amazon Web Services (AWS), положив начало движению облачных вычислений. AWS предоставляет широкий набор сервисов, таких как вычислительные мощности и хранилища данных. Скоро к Amazon.com присоединились Netflix, Microsoft, Google, Apple и IBM, и рынок облачных вычислений разросся. В 2014 году Gartner назвал облачные вычисления среди 10 главных стратегических технологических трендов. «Облако» – это место, где можно хранить данные нужного вам объема, быстро и легко пользоваться сервисами и приложениями. Подключенное к Интернету устройство – единственное, что вам нужно, чтобы воспользоваться программой из «облака» в любое время, в любом месте.

Выделяют три сервисные модели облачных вычислений:

* Infrastructure-as-a-service (IaaS), инфраструктура как услуга, когда потребитель использует вычислительные ресурсы поставщика (сервер, сетевую инфраструктуру, хранилище данных);
* Platform-as-a-service (PaaS), платформа как услуга, когда поставщик предоставляет потребителю доступ к использованию программной платформы;
* Software-as-a-service (SaaS), программное обеспечение как услуга, когда потребитель может пользоваться готовым приложением поставщика.

Задача данного программного проекта заключена в реализации контроля всех подключённых устройств с единого, дружественного пользователю web-интерфейса в реальном времени. Поддержке общения между устройствами и решение ими определенных повседневных задач без участия человека. Предоставление пользователю возможности гибкой настройки системы под свои потребности. Подразумевается, что пользователь выбирает вид управляемых устройств среди поддерживаемых системой в соответствии с задачами, исполняемые ими.

Проект предоставляет возможности по управлению освещением, системой микроклимата, управлению электроприводами, внедрению системы безопасности.

Основные возможности:

* управление различными типами устройств
* веб-доступ с любого устройства в глобальной сети
* редактор сценариев работы устройств
* web-интерфейс с обновлением в реальном времени
* push-уведомления
* интеграция со сторонними веб-сервисами (сервис погоды)
* модель безопасности с разграничением доступа между пользователями
* просмотр статистики работы устройств
* пользователю доступны данные всех устройств без их сохранения в облаке
* синхронизация состояния устройств, управляемых непосредственно и через web-интерфейс
* plug and play(PnP) авто определение новых подключённых устройств

Web-сервис представляет собой сервер-клиентское приложение, в котором сервер находится на облачной PaaS (Platform as a Service) платформе Heroku. Клиентская часть выполняется в браузере. Клиент выполняет запросы на сторонние сервисы, такие как сервис погоды openWeatherMap, используя его API а также устанавливает соединение с домашним сервером автоматизации, через который происходит управление устройствами и от которого приходят уведомления о состоянии системы в реальном времени.

Разработка web-приложения производилась с помощью технологий Ruby on Rails, AngularJS, jQuery, Bootstrap, PatternFly, HTML5. Приложение имеет REST (Representation state transfer) архитектурный стиль. Ruby on Rails реализует паттерн MVC (Model-View-Controller), AngularJS – MVW (Model-View-Whatever).

СПИСОК ЛИТЕРАТУРЫ

[1] Флэнаган Д., Мацумото Ю. Язык программирования Ruby. – СПб.: Питер, 2011. – 496с.

[2] Hartl M., The Ruby On Rails Tutorial Learn web development with Rails [Электронный ресурс]. – Режим доступа: http://railstutorial.ru/.

[3] Руби С., Томас Д., Хэнссон Д. Гибкая разработка веб-приложений в среде Rails. 4-е изд. – СПб.: Питер,2012. – 464 с.: ил.

[4] AngularJS API Reference [Электронный ресурс]. – Режим доступа: https://docs.angularjs.org/api.

[5] Флэнаган Д. JavaScript. Подробное руководство. – Пер. с англ. – СПб: Символ-Плюс,2008. –992 с., ил.

[6] Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб: Питер, 2001. –368 с.: ил.

[7] Coperland D. Powerful, Effective, and Efficient Full-Stack Web Development – Dallas, Texas, 2016.

[8] Coperland D. Powerful, Effective, and Efficient Full-Stack Web Development Second Edition – Raleigh, North Carolina, 2016.

[9] AngularJS with Ruby on Rails [Электронный ресурс]. – Режим доступа: http://angular-rails.com/.