МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет**

**(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

|  |  |
| --- | --- |
| РАБОТА ПРОВЕРЕНА  Рецензент  Генеральный директор  ООО «Системные сети»  \_\_\_\_\_\_\_\_\_\_ С.С. Зайцев  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2019 г. | ДОПУСТИТЬ К ЗАЩИТЕ  Заведующий кафедрой,  д.ф.-м.н., профессор  \_\_\_\_\_\_\_\_\_\_ Л.Б. Соколинский  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

**Реализация прототипа приложения для обработки данных IoT на базе платформ Microsoft Azure и Amazon Web Services**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЮУрГУ – 02.03.02.2019.115-007.ВКР

|  |  |
| --- | --- |
|  | Научный руководитель,  к.ф.-м.н., доцент  \_\_\_\_\_\_\_\_\_\_Г.И. Радченко    Автор работы, студент группы КЭ-401  \_\_\_\_\_\_\_\_\_\_Р.А. Бобин  Ученый секретарь  (нормоконтролер)  \_\_\_\_\_\_\_\_\_\_\_\_\_ О.Н. Иванова  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

Челябинск-2019

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 5](#_Toc1727867)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И СУЩЕСТВУЮЩИХ РАБОТ ПО ТЕМАТИКЕ ДИПЛОМА 8](#_Toc1727868)

[1.1. Концепция интернета вещей и цифровых двойников 8](#_Toc1727869)

[1.2. Концепция облачных технологий 9](#_Toc1727870)

[1.3. Анализ технологий для обработки данных, генерируемых устройствами интернета вещей (сократить) 11](#_Toc1727871)

[2. ТРЕБОВАНИЯ К СИСТЕМЕ 16](#_Toc1727872)

[2.1. Функциональные требования к системе 16](#_Toc1727873)

[2.2. Нефункциональные требования к системе 16](#_Toc1727874)

[2.3. Варианты использования системы 16](#_Toc1727875)

[АРХИТЕКТУРА СИСТЕМЫ 17](#_Toc1727876)

[2.4. Компоненты системы 17](#_Toc1727877)

[2.5. Проектирование реализации прецедентов 17](#_Toc1727878)

[3. РЕАЛИЗАЦИЯ СИСТЕМЫ 18](#_Toc1727879)

[4. ТЕСТИРОВАНИЕ 19](#_Toc1727880)

[4.1. Выбор способов тестирования 19](#_Toc1727881)

[4.2. Описание тестов 19](#_Toc1727882)

[ЗАКЛЮЧЕНИЕ 19](#_Toc1727883)

[СПИСОК ЛИТЕРАТУРЫ 20](#_Toc1727884)

[ПРИЛОЖЕНИЕ 22](#_Toc1727885)

# 

# ВВЕДЕНИЕ

**Актуальность темы**

Концепция Интернета вещей (IoT) не нова, однако особенную популярность она приобрела именно в последние годы. Данная концепция представляет собой вычислительную сеть физических предметов, оснащенных встроенными технологиями для взаимодействия друг с другом или с внешней средой [gartner]. Общий мировой объем капиталовложений в IoT в 2018 году составил 646 миллиардов долларов США. Прогноз на 2019 год — 745 миллиардов долларов США, на 2022 — более 1 триллиона долларов. [idc research]

Развитие распределенной сетевой инфраструктуры в автоматизированных системах управления технологическим процессом привело к появлению Промышленного интернета вещей (IIoT). Одним из преимуществ внедрения этого подхода является возможность создания цифрового двойника разрабатываемой системы (Digital Twin). Цифровой двойник — это иерархическая система математических моделей, вычислительных методов и программного обеспечения, которая обеспечивает синхронизацию между состоянием реально существующего процесса или системы и сопутствующей виртуальной копией. [ucc - 2017]

Благодаря развитию облачных технологий появилась возможность создать инфраструктуру хранения данных, способную поддерживать Интернет вещей. Публичные облачные платформы предоставляют множество решений для цифрового преобразования бизнеса. Гибкость и автоматизированность полученной среды достигается за счет использования прикладных программных интерфейсов (API). Это позволяет различным устройствам и системам взаимодействовать между собой, даже если они работают на основе разных стандартов и протоколов [грингард].

Данная работа направлена на изучение и анализ инструментов, предоставляемых облачными платформами Microsoft Azure и Amazon Web Services для обработки данных IoT.

**Цели и задачи работы**

Целью данной работы является реализация прототипа приложения для обработки данных IoT на базе платформ Microsoft Azure и Amazon Web Services. Для достижения цели работы, необходимо решить следующие задачи:

* изучение инструментов, предоставляемых используемыми платформами;
* анализ литературы и смежных проектов, связанных с мобильной кулинарией;
* определение требований к приложению;
* разработка архитектуры приложения;
* разработка схемы взаимодействия пользователя с интерфейсом приложения;

**Объем и структура работы**

Работа состоит из введения, пяти глав и заключения. Объем работ составляет XX страниц, объем библиографии — XX источников, объем приложения — XX страницы.

**Краткое содержание работы**

В главе «Анализ предметной области и существующих работ по тематике диплома» приведен обзор крупнейших облачных платформ для создания цифровых двойников и литературных источников по тематике работы. В главе «Требования к системе» описаны требования к системе, актеры и варианты использования системы. В главе «Архитектура системы» представлена архитектура системы и ее описание. В главе «Реализация системы» представлены результаты реализации системы и описаны используемые технологии. В главе «Тестирование» представлен отчет о тестировании системы. В заключении представлены результаты работы. В приложении приведено описание диаграмм деятельности системы.

## АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И СУЩЕСТВУЮЩИХ РАБОТ ПО ТЕМАТИКЕ ДИПЛОМА

## Концепция интернета вещей и цифровых двойников

Появление интернета вещей (IoT) представляет новую эру в области вычислительной техники и технологий. Эта концепция безграничные возможности и большие выгоды для науки и общества. Согласно отчетам iot-analytics.com[] уже сейчас в мире насчитывается около 7 миллиардов IoT-устройств, постоянно генерирующих огромное количество данных. И по прогнозам ресурса рынок таких устройств будет стремительно расти. Однако появляется проблема выбора технологий и развертывание фрэймоворков для подобных ресурсоёмких вычислений.

Самым перспективным направление развития IoT является промышленный интернет вещей (IIoT). Принцип работы технологии заключается в следующем: первоначально устанавливаются датчики, исполнительные механизмы, контроллеры и человеко-машинные интерфейсы на ключевые части оборудования, после чего осуществляется сбор информации, которая впоследствии позволяет компании приобрести объективные и точные данные о состоянии предприятия. Обработанные данные доставляются во все отделы предприятия, что помогает наладить взаимодействие между сотрудниками разных подразделений и принимать обоснованные решения. Эффективным способом является создание цифровых двойников [про цифр дв]. В статье [ucc] представлены основные концепции системы цифровых двойников:

* (отклик) в реальном времени;
* наличие системной иерархии;
* самосовершенствование;
* гибкость.

Можно написать еще что-нибудь.

## Концепция облачных технологий

Революционное распространение IoT-устройств создает большие про-блемы, такие как выбор и развертывание фрэймоворков для подобных ресурсоёмких вычислений. Для получения и обработки данных целесообразно использовать облачные платформы, которые, в свою очередь, стремительно развиваются и увеличивают набор инструментов, доступный для использования в самых различных областях.

Облачные платформа основаны на технологии облачных вычислений — модели обеспечения удобного сетевого доступа по требованию к некоторому общему фонду конфигурируемых вычислительных ресурсов (например, сетям передачи данных, серверам, устройствам хранения данных, приложениям и сервисам — как вместе, так и по отдельности), которые могут быть оперативно предоставлены и освобождены с минимальными эксплуатационными затратами или обращениями к провайдеру. [NIST]

*Можно написать что-нибудь про индустрию 4.0. [deloitte]*

## Анализ технологий для обработки данных, генерируемых устройствами интернета вещей (сократить)

Вышеописанные платформы предоставляют самые разные службы и решения, которые помогают создавать приложения IoT. Рассмотрим

1. Azure Digital Twins от Microsoft. Azure Digital Twins представляет собой службу Интернета вещей, с помощью которой можно создать комплексные модели физического окружения. С ее помощью можно создавать пространственные интеллектуальные графы для моделирования связей и взаимодействий между людьми, пространствами и устройствами. Azure Digital Twins позволяет запрашивать данные из физического пространства, а не из многих разрозненных датчиков. С помощью этой службы можно создавать многоразовые, высокомасштабируемые, пространственно-ориентированные возможности, которые связывают потоковые данные в цифровом и реальном мире.

Azure Digital Twins имеет следующие ключевые возможности.

Пространственный интеллектуальный граф

Пространственный интеллектуальный граф (или пространственный граф) является виртуальным представлением физического окружения. С его помощью можно моделировать связи между пользователями, расположениями и устройствами.

Цифровые модели объектов двойников

Цифровые модели объектов двойников — это предопределенные протоколы устройств и схема данных. Они представляют потребности определенной предметной области решения, чтобы ускорить и упростить разработку.

Множественные и вложенные клиенты

Решения безопасно масштабируются и поддерживается их повторное использование для нескольких клиентов. Также можно создать несколько вложенных клиентов, к которым можно получить доступ и которые можно использовать изолированным и безопасным образом.

Расширенные вычислительные возможности

С помощью определяемых пользователем функций можно определять и запускать пользовательские функции в отношении входящих данных устройства для отправки сигналов в заранее определенные конечные точки. Эта расширенная возможность улучшает процессы настройки и автоматизацию задач устройства.

Встроенное управление доступом

Используя функции управления доступом и идентификацией, такие как управление доступом на основе ролей и Azure Active Directory, можно безопасно контролировать доступ отдельных лиц и устройств.

[azure-doc] + [схема]

1. AWS IoT Core. AWS IoT Core – это управляемый облачный сервис, который позволяет подключенным устройствам просто и безопасно взаимодействовать с облачными приложениями и другими устройствами, а также без труда создавать приложения для сбора, обработки и анализа данных, генерируемых подключенными устройствами, а также для выполнения действий на основе этих данных без необходимости управлять какой‑либо инфраструктурой. AWS IoT Core поддерживает протоколы HTTP, WebSockets и упрощенный протокол связи MQTT, специально спроектированный для поддержки нестабильных подключений, сокращения объема кода, передаваемого устройству, и для работы в сетях с низкой пропускной способностью.

Основные возможности.

SDK AWS IoT для устройств.

Пакет SDK AWS IoT для устройств позволяет просто и быстро подключить аппаратное устройство или мобильное приложение к AWS IoT Core. SDK AWS IoT для устройств обеспечивает подключение и аутентификацию устройств, а также обмен сообщениями с платформой AWS IoT Core по протоколам MQTT, HTTP или WebSockets. Пакет SDK AWS IoT для устройств поддерживает языки C, JavaScript и Arduino и включает в себя клиентские библиотеки, руководство для разработчиков и руководство по переносу для производителей.

Шлюз устройств.

Шлюз устройств служит точкой входа для устройств IoT, подключающихся к AWS. Шлюз устройств управляет всеми активными подключениями устройства и реализует семантику различных протоколов, чтобы обеспечить надежную и эффективную связь устройств с AWS IoT Core.

Message Broker.

Message Broker – это высокопроизводительный брокер сообщений, работающий по стандарту «издатель-подписчик», который надежно и с низкой задержкой передает сообщения всех ваших устройств IoT и приложений в нужном направлении. Гибкий характер структуры тем Message Broker позволяет получать сообщения с любого необходимого количества устройств или отправлять любое количество сообщений на устройства.

Аутентификация и авторизация.

Платформа AWS IoT Core обеспечивает взаимную аутентификацию и шифрование во всех точках подключения. Таким образом, любой обмен данными между устройствами и AWS IoT Core происходит только после подтверждения идентификации. AWS IoT Core поддерживает метод аутентификации AWS (называемый SigV4), аутентификацию на основе сертификата X.509 и специальную аутентификацию на основе токенов (через настраиваемые модули авторизации).

Реестр.

Реестр устанавливает идентификацию для устройств и позволяет отслеживать метаданные, такие как атрибуты или возможности устройства. Реестр позволяет уникальным образом идентифицировать каждое устройство в соответствии с единым форматом, не зависящим от типа устройства или его подключения. Он также поддерживает использование метаданных, описывающих возможности устройства.

Тени устройств.

С помощью AWS IoT Core можно создать постоянную виртуальную версию каждого устройства, так называемую тень устройства, содержащую его последнее состояние и позволяющую приложениям или другим устройствам считывать сообщения и взаимодействовать с данным устройством. Тени устройств хранят последнее зарегистрированное состояние и желаемое будущее состояние каждого устройства, даже если оно находится в автономном режиме. Получить последнее зарегистрированное состояние устройства или задать для него требуемое будущее состояние можно с помощью API или сервиса правил. Доступ приложений к теням осуществляется с помощью доступных REST API.

Сервис правил.

Сервис правил позволяет создавать приложения IoT для сбора, обработки и анализа данных, генерируемых подключенными устройствами, и выполнения действий с ними в глобальных масштабах без необходимости управления какой-либо инфраструктурой. Сервис правил оценивает входящие сообщения, публикуемые в AWS IoT Core, а затем преобразует и доставляет их другому устройству или облачному сервису с учетом заданных бизнес-правил. Правило можно применять к данным от одного или от многих устройств и выполнять на его основе одно действие или же множество параллельных действий.

[aws-doc] + [схема]

*Вывод по сравнению платформ.*

Обе платформы определяют только модель данных. Без действий и событий. Форматы сериализации обеих систем невзаимозаменяемы. Сервисы также не определяют привязку (binding) протоколов. Производителям устройств нужно подстраиваться под каждую среду. Это может превышать возможности очень маленьких устройств.

[w3]. Отсюда можно взять краткую сводку и некоторые схемы. А также модель oracle.

**Вывод**

Анализ существующих инструментов для создания приложений по обработке данных показывает, что на базе облачных платформ возможно создание качественных IoT-приложений. Это говорит о том, что решение данной задачи является возможным и актуальным.

## ТРЕБОВАНИЯ К СИСТЕМЕ

## Функциональные требования к системе

## Нефункциональные требования к системе

## Варианты использования системы

## АРХИТЕКТУРА СИСТЕМЫ

## Компоненты системы

## Проектирование реализации прецедентов

## РЕАЛИЗАЦИЯ СИСТЕМЫ

## ТЕСТИРОВАНИЕ

## Выбор способов тестирования

## Описание тестов

# ЗАКЛЮЧЕНИЕ

# СПИСОК ЛИТЕРАТУРЫ

1. Interface Builder. [Электронный ресурс] URL: https://developer.apple.com/xcode/interface-builder/ (дата обращения: 20.04.2018).
2. iOS Databases. [Электронный ресурс] URL: https://rollout.io/blog/ios-databases-sqllite-core-data-realm/ (дата обращения: 20.04.2018).
3. Nimbalkar R.R. Mobile Application Testing And Challеnges. // International Journal of Science and Research, 2013. – Vol. 2. – P. 56 – 58.
4. Suhailah Mohd Yusof, Sharifah Fateen Syuhada Syed Lokman. Personal Financial Planner: A Mobile Application that Implementing Forward Chaining Technique for Notification Mechanism. // IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), April, 2014. – P. 65 – 69.
5. Tarasewich P. Designing Mobile Commerce Applications. // Communications of the acm. December 2003. Vol. 46. Issue 12. – P. 57 – 60.
6. Арлоу Д., Нейштадт А. UML 2 и Унифицированный процесс:.Практический объектно-ориентированный анализ и проектирование (пер. с англ. Шатохиной Н.). – 2-е изд. – М.: Символ Плюс, 2008. – 622 с.
7. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования; [пер. с англ.: А. Слинкин науч. ред.: Н. Шалаев]. Хелм Р., Джонсон Р., Влиссидес Д. СПб: Питер, 2014. – 366 с.
8. Головач В.В. Дизайн пользовательского интерфейса v1.2. [Электронный ресурс] URL: http://uibook2.usethics.ru/ (дата обращения: 21.04.2018).
9. Документация Apple iOS. [Электронный ресурс] URL: https://developer.apple.com/library/ios/documentation (дата обращения: 19.04.2018).
10. Кришна Г. Хороший интерфейс – невидимый интерфейс. – СПб: Питер, 2016. – 256 с.
11. Марк Д. iOS 6 SDK. Разработка приложений для iPhone, iPad и iPod touch на Objective-C в Xcode. Наттинг Д., Ламарш Д., Олссон Ф. СПб.:Вильямс, 2013. – 672 с
12. Официальная документация платформы Realm. [Электронный ресурс] URL: https://realm.io/docs/objc/latest/ (дата обращения: 20.04.2018).
13. Поваренок.ру. [Электронный ресурс] URL: http://www.povarenok.ru/ (дата обращения: 20.04.2018).
14. Применение специй в кулинарии. [Электронный ресурс] URL: https://vsadu.ru/post/kakie-specii-ispolzovat-v-prigotovlenii.html (дата обращения: 20.04.2018).
15. Статистика мобильных приложений в AppStore. [Электронный ресурс] URL: https://www.theverge.com/2016/6/13/11922926/apple-apps-2-million-wwdc-2016 (дата обращения: 19.04.2018).

# ПРИЛОЖЕНИЕ

Описание диаграмм деятельности системы Appetizer.

**Табл. 1.** Описание диаграммы деятельности прецедента «Подобрать сочетаемый продукт»

|  |
| --- |
| Прецедент «Подобрать сочетаемый продукт» |
| ID: 1 |
| Краткое описание:  Подбор сочетаемых продуктов к выбранному продукту. |
| Главные актеры:  Пользователь |
| Второстепенные актеры:  Нет |
| Предусловия:  Нет. |
| Основной поток:   1. Прецедент начинается на экране «Подобрать сочетания»»; 2. Пользователь выбирает категорию продукта, для которого он хочет подобрать сочетание; 3. Система переводит пользователя на экран продуктов выбранной категории. Из предложенного списка, пользователь выбирает свой главный продукт; 4. Открывается окно, в котором нужно выбрать категорию продуктов для подбора сочетания. После выбора нужной категории, пользователь переходит к следующему шагу; 5. Пользователь выбирает понравившийся продукт из предложенного списка; 6. Открывается окно с выбранным сочетанием; 7. Если пользователю нужно подобрать еще сочетания, система переводи пользователя на шаг 3, иначе – работа с данным прецедентом закончена. |
| Постусловия:  На экране отображается подобранное пользователем сочетание. |
| Альтернативные потоки:  Нет. |

**Табл. 2.** Описание диаграммы деятельности прецедента «Добавить продукты в список покупок»

|  |
| --- |
| Прецедент «Добавить продукты в список покупок» |
| ID: 2 |
| Краткое описание:  Добавление продуктов из сочетания в список покупок. |
| Главные актеры:  Пользователь |
| Второстепенные актеры:  Нет |
| Предусловия:  Создание подобранного сочетания продуктов. |
| Основной поток:   1. Прецедент начинается на экране с подобранным сочетанием; 2. Пользователь нажимает на кнопку «Добавить в список покупок»; 3. Система открывает окно для ввода названия списка продуктов; 4. Пользователь нажимает кнопку «Сохранить». |
| Постусловия:  Система автоматически добавляет все продукты из сочетания в список покупок с возможностью отмечать уже имеющиеся продукты. |
| Альтернативные потоки:  Нет. |

**Табл. 3.** Описание диаграммы деятельности прецедента «Сохранить подобранное сочетание»

|  |
| --- |
| Прецедент «Сохранить подобранное сочетание |
| ID: 3 |
| Краткое описание:  Сохранение подобранного сочетания в системе. |
| Главные актеры:  Пользователь |
| Второстепенные актеры:  Нет |
| Предусловия:  Создание подобранного сочетания продуктов. |
| Основной поток:   1. Прецедент начинается на экране с подобранным сочетанием; 2. Пользователь нажимает на кнопку «Сохранить сочетание»; 3. Система открывает окно для ввода названия сочетания; 4. Пользователь нажимает кнопку «Сохранить». |
| Постусловия:  Система автоматически сохраняет подобранное пользователем сочетание. |
| Альтернативные потоки:  Нет. |

**Табл. 4.** Описание диаграммы деятельности прецедента «Добавить новый продукт»

|  |
| --- |
| Прецедент «Добавить новый продукт» |
| ID: 4 |
| Краткое описание:  Добавление нового продукта и его сочетания пользователем. |
| Главные актеры:  Пользователь |
| Второстепенные актеры:  Нет |
| Предусловия:  Нет |
| Основной поток:   1. На экране выбора продуктов нажать на кнопку «Добавить»; 2. Ввести название продукта; 3. Выбрать картинку для продукта; 4. Добавить к новому продукту сочетаемые продукты; 5. Если выбраны все продукты, сочетаемые с новым, то работа с прецедентом заканчивается; 6. Иначе, пользователь возвращается к пункту 4. |
| Постусловия:  Система добавляет новый продукт и его сочетания в базу данных для дальнейшей работы пользователя с приложением. |
| Альтернативные потоки:  Нет. |

**Табл. 5.** Описание диаграммы деятельности прецедента «Добавить новое сочетание»

|  |
| --- |
| Прецедент «Добавить новое сочетание» |
| ID: 5 |
| Краткое описание:  Сохранение подобранного сочетания в системе. |
| Главные актеры:  Пользователь |
| Второстепенные актеры:  Нет |
| Предусловия:  Нет |
| Основной поток:   1. На экране выбранной категории пользователь нажимает на кнопку «Изменить»; 2. Пользователь выбирает продукт, для которого хочет добавить новое сочетание 3. Пользователь выбирает новый продукт для сочетания из списка; 4. Если выбраны все продукты, то работа с прецедентом заканчивается; 5. Иначе, пользователь возвращается к пункту 3. |
| Постусловия:  Система обновляет информацию о сочетаниях выбранного продукта. |
| Альтернативные потоки:  Нет. |