Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт
Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О.В.Непомнящий
подпись инициалы, фамилия
« ___» _____ 2018 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Голосовое управление системой «Умный дом»
тема

Руководитель	_____ подпись, дата	доцент, канд. техн. наук должность, ученая степень	Н.Ю. Сиротинина инициалы, фамилия
Выпускник	_____ подпись, дата		С.Г. Алимовнин инициалы, фамилия
Нормоконтролер	_____ подпись, дата	доцент, канд. техн. наук должность, ученая степень	В.И. Иванов инициалы, фамилия

Красноярск 2018

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт
Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О.В.Непомнящий
подпись инициалы, фамилия
« ___» _____ 20__ г.

ЗАДАНИЕ

НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

в форме бакалаврской работы

Студенту Алимовнину Семёну Геннадьевичу

фамилия, имя, отчество

Группа КИ14-07Б Направление (специальность) 09.03.01

номер код

Информатика и вычислительная техника

наименование

Тема выпускной квалификационной работы Голосовое управление системой «Умный дом».

Утверждена приказом по университету № _____от _____

Руководитель ВКР Н.Ю. Сиротинина, доцент, канд. техн. наук.

инициалы, фамилия, должность, учёное звание и место работы

Исходные данные к работе: Объект проектирования: голосовое управление системой «Умный дом». Исходные данные: описание основных функций системы. Особые требования: надежность, безопасность, адаптивность, ориентация на потребности лиц с ограниченными возможностями здоровья. Информационные материалы: документация на используемое оборудование и программное обеспечение; учебные издания; информационные ресурсы библиотеки СФУ. Требования к оформлению работы: согласно стандарту организации «Система менеджмента качества – Общие требования к построению, изложению и оформлению документов учебной деятельности» СТО 4.2–07–2014.

Руководитель ВКР _____ Н.Ю. Сиротинина

подпись инициалы, фамилия

Задание принял к исполнению _____ С.Г. Алимовнин

подпись инициалы, фамилия

«___» _____ 20__г

РЕФЕРАТ

Выпускная квалификационная работа по теме «Голосовое управление системой «Умный дом»» содержит в себе 50 страниц текстового документа, 1 приложение, 28 использованных источников, 14 иллюстраций, 1 таблица.

ГОЛОСОВОЕ УПРАВЛЕНИЕ, МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, УМНЫЙ ДОМ

Цель работы: разработка мобильного приложения для платформы Android с функцией голосового управления системой «Умный дом».

Задачи:

- анализ существующих аналогов интерфейса голосового управления;

- сбор и анализ информации о существующих технологиях голосового распознавания и взаимодействия с «Умным домом»;

- проектирование архитектуры и интерфейса мобильного приложения, удовлетворяющие общим стандартам платформы Android;

- реализация приложения «MyHome».

Во введении раскрывается актуальность работы, изложены цели и задачи.

В первой главе произведен обзор предметной области и анализ аналогов разрабатываемого приложения.

Во второй главе разработаны функции и подсистемы мобильного приложения, выбраны API для голосового распознавания и протокол обмена данными.

В третьей главе описывается реализация мобильного приложения и его подсистем.

В четвертой главе описывается пользовательский интерфейс для пользователей, а также методы тестирования приложения.

СОДЕЖАНИЕ

ВВЕДЕНИЕ

Система «Умный дом» – это сложная концепция, которая позволяет объединить разнообразные устройства домашней автоматизации в единый комплекс. Важной частью системы «Умный дом» (УД) является пользовательский интерфейс. От его реализации зависит удобство использования всей системы в целом. Есть категория пользователей, которая в последнее время часто становится целевой при разработке УД. Это пользователи с ограниченными возможностями здоровья (ОВЗ). Для людей с ОВЗ особенно важна именно реализация интерфейса.

Если пользовательский интерфейс выделен, как отдельная подсистема УД, он может быть адаптирован к индивидуальным потребностям пользователя с минимальной доработкой прочих подсистем. В соответствии с потребностями пользователя может использоваться управление жестами, датчиками нейронных импульсов, управление голосом и т.д. Современная техника позволяет реализовать такой вариант интерфейса, как голосовой. Данный вариант необходим пользователям с нарушениями зрения, двигательных функций. Помимо этого, он во многих ситуациях может быть востребован пользователями без ограничений здоровья. Голосовой интерфейс позволяет отказаться от ввода информации вручную, с помощью кнопок управления или клавиатуры, а также от восприятия ответных сообщений в текстовом виде. В то же время, он менее точен, так как на вводимую информацию могут повлиять внешние шумы.

Данное направление в настоящее время активно развивается. Начиная с 2014 года начали появляться голосовые помощники. Google и Amazon уже презентовали широкой общественности Google Home и Alexa. Представленные в виде колонок, помощники реагируют на обширный диапазон голосовых команд, и могут исполнять роль посредника при управлении системами домашней автоматизации в том числе.

Целью выпускной квалификационной работы является разработка мобильного приложения для платформы Android с функцией голосового управления системой «Умный дом».

В ходе разработки должны были решены следующие задачи:

- анализ существующих аналогов интерфейса голосового управления;

- сбор и анализ информации о существующих технологиях голосового распознавания и взаимодействия с «Умным домом»;

- проектирование архитектуры и интерфейса мобильного приложения, удовлетворяющие общим стандартам платформы Android;

- реализация приложения «MyHome».

В результате реализовано мобильное приложение, включающее в себя функции голосового управления системой «Умный дом».

Данное приложение было протестировано в реальных условиях и подтвердило свою работоспособность. Результаты работы апробированы в двух публикациях в материалах конференций, получены дипломы 1 степени.

1 Сравнительный анализ реализаций голосовых интерфейсов

1.1 Обзор предметной области

Рассматривая систему «Умный дом», можно выделить несколько типов интерфейсов для взаимодействия с пользователем:

- кнопочный или сенсорный интерфейс в виде пульта управления;
- сенсорная панель со встроенным программным обеспечением;
- приложение, устанавливаемое на пользовательский персональный компьютер или смартфон.

Сенсорная панель является не очень удобным интерфейсом, так как появляется необходимость устанавливать его в каждой комнате для удобства пользования, иначе необходимо переходить в ту комнату, где он установлен.

Пульт управления также не очень хорош во взаимодействии, т.к. лишен функций удаленного управления «Умным домом», например, возможности дистанционно его выключить или снять охрану.

Рассматривая приложение также можно выделить 2 типа реализации:

- экранный интерфейс, где пользователь должен нажать на определенную пиктограмму или ввести текст для выполнения действия;
- голосовой интерфейс, который позволяет распознавать голосовой текст и на основе результатов выполнять необходимые действия.

Голосовой интерфейс – это программный продукт, который при помощи голосовой или речевой платформы позволяет взаимодействовать пользователю с системой «Умный дом», отдавая команды для запуска отдельных сценариев и действий голосом и получая информацию в форме звуковых сообщений. Соответственно, основными специфическими функциями таких интерфейсов являются распознавание голосовой команды и синтез звуковых сообщений [2].

Ранее контролировать устройство при помощи голоса было возможно только в научной фантастике. С развитием технологий голосовой интерфейс стал всё более распространённым, человек всё чаще пользуется преимуществами этой бесконтактной технологии [3].

Голосовые интерфейсы удобны, когда вводить текст сложно или неудобно. Например, во время вождения автомобиля пользователь может проговорить свой запрос, продиктовать нужный адрес, проверить пробки в приложении навигатора. Или же если пользователь выполняет слишком много задач и не может сконцентрироваться на одной.

Возможным решением реализации голосового интерфейса внутри помещения могло бы быть размещение микрофонов и источников звука в различных точках помещения, таким образом, чтобы пользователь мог в любой момент произнести команду и услышать ответ. Однако попытки реализовать такие системы столкнулись с тем, что потребовалось большое количество чувствительных микрофонов, поэтому система становилась технически сложной и дорогостоящей. В то же время, современный человек практически постоянно носит с собой мобильный телефон, это делает реализацию голосового интерфейса намного дешевле, а также добавляет возможность удаленного контроля. Поэтому реализация мобильного приложения является более удобным решением для голосового управления «Умным домом».

Проект по разработке мобильного приложения направлен именно на голосовое управление «Умным домом»,

1.2 Критерии сравнения систем голосового управления

Системы голосового управления, используемые в умном доме, должны соответствовать следующим критериям:

- поддержка русского языка;
- создание собственных команд;

- возможность обратного взаимодействия с пользователем;
- стоимость.

1.3 Обзор аналогов разрабатываемой системы

1.3.1 Семейство Amazon Echo

Amazon Echo – серия «умных» устройства, разработанных компанией Amazon. Первое поколение устройств, из данной линейки, появилось в 2014 году с выходом Bluetooth-колонки Amazon Echo. Amazon Echo – устройства, созданные как голосовые помощники, работающие в режиме “AlwaysOn”. Взаимодействие с ними происходит только голосом, без использования кнопок. В колонках используется собственная технология распознавания речи Alexa. Данная линейка устройств может управлять устройствами умного дома, воспроизводить музыку с телефона, Amazon Prime или Spotify, а также заказывать продукты с Amazon [4].

В Echo второго поколения заявлено лучшее звуковоспроизведения благодаря системе Dolby, которая была представлена летом 2016 года – это позволяет использовать мультикомнатную аудиосистему.

В гаджет Amazon Echo Show добавлен тачскрин, камера и умный спикер, позволяющий делать видеозвонки, просматривать видеокамеры, смотреть Amazon Prime Video.

На 2018 год компания свыше 10 устройств для умного дома. Цена стартует с 36$.

Преимущества:

- возможность голосового оповещения с использованием колонок.

Недостатки:

- не имеет поддержки русского языка;
- для создания собственных команд необходимо знание программирования;
- высокая стоимость.

1.3.2 Google Home

Google Home – беспроводной динамик, снабженный голосовым управлением, разработанный в Google. Продукт был объявлен 18 мая 2016 года на Google I/O. Представляет собой первый смарт-динамик Google. Также стал одним из устройств, поддерживающих работу персонального ассистента «Google Assistant», наряду с текстовым чатом Allo и видеочатом Duo [5].

Продукт очень похож по своей концепции на Amazon Echo, и является его прямым конкурентом в промышленности «умных» колонок и персональных помощников. Также часть функциональности похожа на голосовой помощник Apple Siri [6].

Стоимость от 99$.

Преимущества:

- легкость интеграции в существующие решения, из-за поддержки разработчиков сторонних модулей.

Недостатки:

- высокая стоимость.

1.3.3 Apple Siri

Apple Siri – облачный персональный помощник и вопросно-ответная система, программный клиент которой входит в состав iOS, watchOS, macOS, и tvOS компании Apple. Данное приложение использует обработку естественной речи, чтобы отвечать на вопросы и давать рекомендации. Siri приспосабливается к каждому пользователю индивидуально, изучая его предпочтения в течение долгого времени [7].

Первоначально Siri стало доступно в App Store как приложение для iOS от Siri Inc. Вскоре, 28 апреля 2010 года, Siri Inc. была приобретена Apple Inc. Но ещё до того, как Apple купила Siri, было объявлено, что их программное обеспечение

будет доступно для телефонов BlackBerry и телефонов под управлением Android, затем эти планы были отменены из-за покупки.

Для взаимодействия Siri и «Умного дома» компания Apple создала проект HomeKit [8].

Данная технология обладает следующими возможностями:

- siri знает, какие аксессуары HomeKit настроены в программе «Дом», а также их статус. Siri идентифицирует аксессуары по их именам, местоположению и, по другим сведениям, указанным для них в программе «Дом». Если настроен HomePod, Apple TV или iPad в качестве домашнего центра, то можно использовать Siri для управления домом в любом месте;

- siri может включать и выключать аксессуары HomeKit: от освещения до бытовых приборов;

- siri может регулировать работу аксессуаров HomeKit, например световых приборов или термостата;

- если аксессуары организованы по комнатам или зонам, можно управлять областями дома одной командой;

- при использовании HomePod есть возможность управлять всеми аксессуарами HomeKit в комнате, где находится это устройство, одной командой;

- сценарии позволяют управлять несколькими аксессуарами одновременно. С помощью Siri можно настроить сценарий, используя только голосовые команды;

- проверка состояния дома;

- преимущества данной технологии;

- начиная с версии iOS 8.3 имеется поддержка русского языка;

- бесплатен для использования.

Недостатки:

- необходимо использовать только устройства от компании Apple;

- для работы возможно использовать только сертифицированные компанией Apple компоненты для работы «Умного дома».

1.4 Вывод

Данные реализации являются закрытыми и дорогими для интеграции в «Умный дом», поэтому было принято решение о создании собственной доступной и дешевой реализации голосового интерфейса, который взаимодействует с уже существующей реализацией «Умного дома».

2 Разработка приложения

2.1 Функции и подсистемы приложения

В данном мобильном приложении будут реализованы следующие функции:

- функция автоматического старта приложения при старте операционной системы;
- функция распознавания голосовых команд;
- функция обратной связи с пользователем в виде синтеза голоса или вывода сообщений на экран;
- функция соединения и управления «Умным домом»;
- функция настройки интерфейса приложения;
- функция администрирования приложения.

Функция распознавания голосовых команд является основной функцией мобильного приложения. Функция настройки интерфейса приложения заключает в себя функционал графической настройки приложения с помощью пунктов меню. Функция администрирования приложения позволяет производить первоначальную настройку приложения и дальнейшего изменения параметров.

В мобильном приложении были выделены следующие подсистемы:

- голосовой интерфейс;
- интерфейс к аппаратной части, включающий в себя функцию соединения с «Умным домом»;
- хранение данных;
- подсистема выполнения команд;
- пользовательский интерфейс.

В результате анализа функций предложена следующая структурная схема, изображенная на рисунке 1.

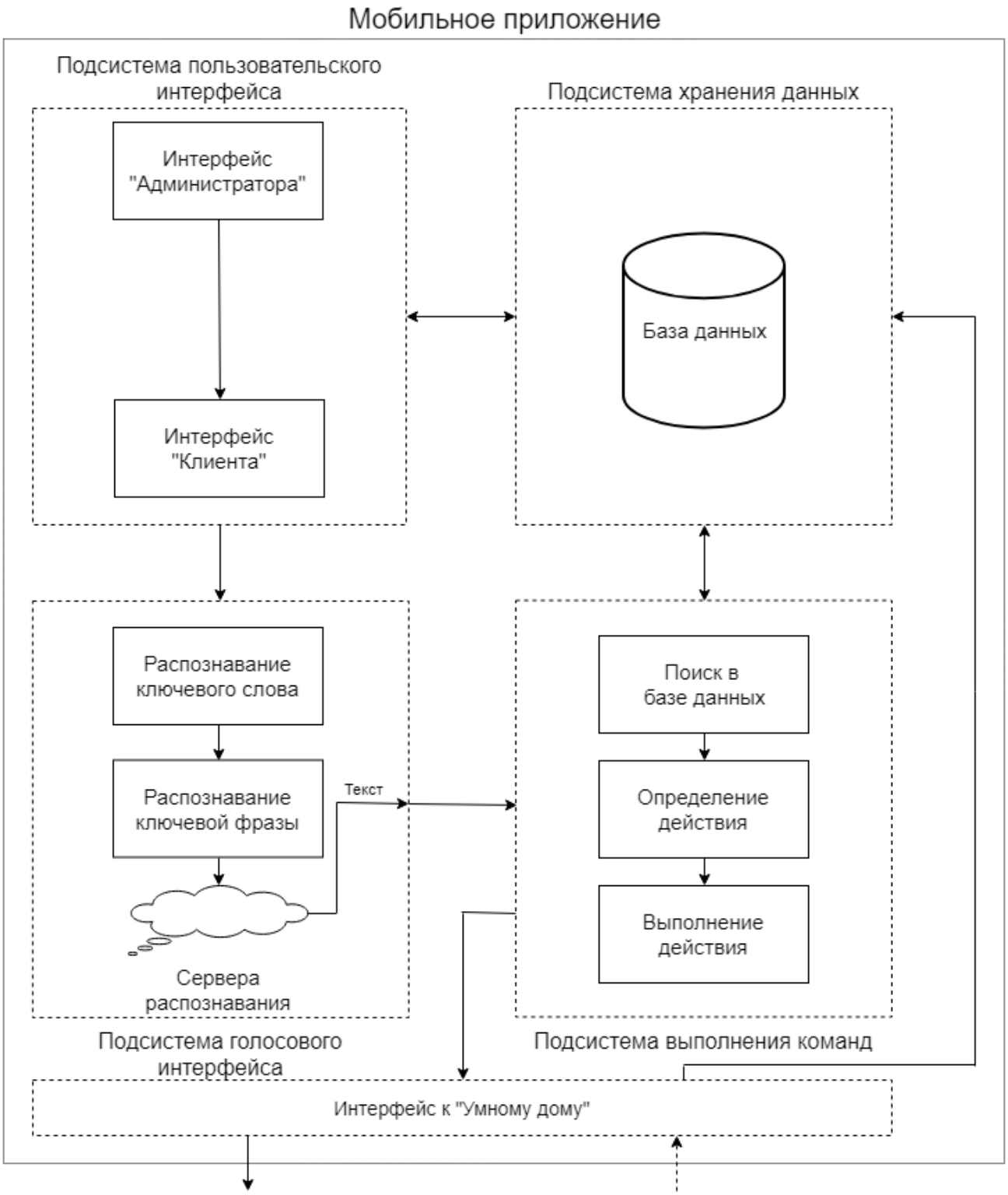

Рисунок 1 – Структурная схема приложения

2.2 Подсистема голосового интерфейса

Реализовать самостоятельно распознавание речи не имеет смысла, потому что на рынке уже существуют системы, открытые для программистов. Данные системы используют быстрые алгоритмы и нейронные сети для анализа, поэтому выбираем из уже существующих [9].

Для реализации функции распознавания голосовых команд нам необходимо проанализировать существующие технологии распознавания речи и выбрать среди них те, которые будут удовлетворять перечисленным критериям.

2.2.1 Анализ технологий распознавания речи

Имеется два подхода к распознаванию речи [10].

Первый заключается в онлайн распознавании голоса. Такие фирмы как Google. Яндекс, Apple, Samsung, Amazon и другие, постоянно совершенствуют свои системы голосового управления умным домом. Эти системы чаще всего используют для своей работы искусственный интеллект: введённая фраза передаётся на центральный сервер, где обрабатывается и отсылается пользователю. Таким образом пользователь получает результат. Такой подход требует наличие постоянного Интернет-соединения [11].

Второй подход основан на сравнении введённых слов с имеющейся базой. Список слов формируется программистом при настройке системы умный дом и учитывает конкретные пожелания владельца. В такой системе нужно чётко формулировать свои запросы, так как система не способна их интерпретировать. Например, включить свет в зале, выключить телевизор, закрыть шторы и т.д. Минусом такого подхода будет сложность в добавлении функционала без знаний языков программирования. Система голосового распознавания тесно связана с системой голосового вывода, так как распознанные команды желательно куда-то выводить. Это можно делать при помощи вывода информации на дисплей, синтеза речи или специально записанных семплов. В системе с онлайн

распознаванием актуально использовать синтез речи или дисплей с текстовой информацией, так как заранее не всегда известно какой результат будет возвращён в ответ на голосовой запрос. В системе, использующей заранее заготовленную базу слов лучше использовать записанные семплы. В такой системе при программировании можно связать голосовой запрос и голосовое сообщение, выводящееся пользователю. Причём в этом случае сообщения будут всегда одинаковыми. Такой подход позволяет создать стабильную и быстро работающую систему, но на разработку этой системы уйдёт много времени [12].

2.2.2 Критерии оценки

При исследовании API для технологий распознавания голоса упор был поставлен прежде всего на: наличие русского языка, корректность результатов распознавания и цену продукта.

Под корректностью подразумеваем совпадение результатов распознавания с тестовым набором сообщений. Набор сообщений включает в себя фразы: «Привет, дом», «Включи свет», «Температура». Данные фразы были повторены несколько раз.

Для выбора технологии были проведены тесты, включающие в себя написание тестовой программы и проведения измерений точности и времени распознавания.

Были рассмотрены следующие реализации.

2.2.3 Речевые технологии SpeechKit

Комплекс речевых технологий от российской компании Яндекс, который включает распознавание и синтез речи, голосовую активацию и выделение смысловых объектов в произносимом тексте. Включает в себя Mobile SDK.

SpeechKit Mobile SDK позволяет встроить распознавание и синтез речи, а также голосовую активацию Яндекса в ваше мобильное приложение для iOS, Android или Windows Phone.

Если количество голосовых обращений к приложению не превышает 10 000 в сутки, использовать SpeechKit Mobile SDK можно бесплатно (базовая версия). Если запросов окажется больше, есть возможность либо оплачивать превышение лимита, либо перейти на тариф бизнес, в котором нет таких ограничений и доступны дополнительные функции – например, создание специальных тематических моделей и уникальных голосов. Бесплатно для образовательных целей [13].

Преимущества:

- поддержка русского языка;
- безопасность: все сообщения передаются в шифрованном виде;
- работает на основных мобильных платформах.

Недостатки:

- возможны ограничения в работе программных продуктов из-за возможного ограничения количества обращений;
- возможные перебои в работе из-за того, что технология еще в разработке.

2.2.4 Google Cloud Speech API

Технология распознавания голоса от компании Google.

Google Cloud Speech API позволяет разработчикам преобразовывать аудио информацию в текст, применяя мощные модели нейронных сетей в простой для использования API [14].

Имеется поддержка множества устройств и программных продуктов. Имеется возможность работы на большинстве современных операционных системах: Android, iOS, Windows 10.

Ценовая политика - 0-60 минут в месяц бесплатно, далее 0,006 $ за 15 секунд речи. Каждый запрос округляется к цифре кратной 15. Первые два месяца бесплатно [15].

Google Cloud Speech API поддерживает более 110 языков (в том числе русский) [16]. Использование нейронных сетей позволяет быстро (порядка 1-2 секунд) преобразовывать голосовую модель в текст.

Преимущества:

- бесплатен для использования на платформе Android;

- быстрое распознавание речи и обработка результатов;

- имеется поддержка русского языка;

- возможность распознавания без использования интернета на платформе Android [17].

Недостатки:

- при использовании на платформе отличной от Android вводится плата за обращения к серверу;

- закрытость платформы.

2.2.5 Alexa Voice Service

Технология распознавания голоса от компании Amazon.

Alexa Voice Service (AVS) позволяет интегрировать голосового помощника Alexa непосредственно в программные продукты. Amazon предоставляем доступ к набору ресурсов для быстрого и легкого создания продуктов с поддержкой Alexa, включая API, инструменты для разработки аппаратного и программного обеспечения и документацию. С AVS есть возможность добавить новый интеллектуальный интерфейс к своим продуктам и предложить клиентам доступ к растущему числу функций Alexa и интеграции в «Умные дома» [18].

Преимущества:

- быстрое распознавание и отклик из-за использования серверов по всему миру.

Недостатки:

- не имеет поддержки русского языка;

- нет бесплатного использования.

2.2.6 SnowBoy

Технология распознавания голоса с открытым исходным кодом под Android, iOS, Windows, JavaScript [19].

Данная технология позволяет запускать распознавание речи прямо на устройстве без использования сторонних серверов. SnowBoy использует языковые модели. Для начала распознавания необходимо записать голос человека, который произносит необходимую фразу для распознавания, из-за этого отпадает возможность

Преимущества:

- запуск распознавания без использования серверов, что позволяет постоянно использовать распознавание на устройстве;
- бесплатная технология.

Недостатки:

- требование составления готовых языковых моделей.

2.2.7 Тестирование систем распознавания речи

Поскольку надежность распознавания речи является важной характеристикой, для выбора реализации API было проведено тестирование. Для это была реализована тестовая программа, в которую включаются функции:

- взаимодействия с API;
- замера времени распознавания от запуска до конечного результата в виде текста.

Для трех API, «Речевые технологии SpeechKit», «Google Cloud Speech API» и «SnowBoy», были проведены в общей сумме тридцать тестов. Результаты тестирования были записаны в таблицу 1.

Таблица 1 – Результаты тестирования

Название технологии	**Среднее время распознавания после 10 запусков (сек)**	**Точность распознавания после 10 запусков**
Речевые технологии SpeechKit	3.9	Неверные результаты полностью несовпадающие с изначальным текстом
Google Cloud Speech API	1.6	8 из 10 правильно определенных текстов
SnowBoy	1.2	7 из 10 правильно определенных текстов
Технология Alexa Voice Service не была включена в тестирование, так как не имеет поддержки русского языка, что нарушает критерии исследования.		

В результате тестирования было получено решение, что SnowBoy наиболее подходит для распознавания ключевого слова, а Google Cloud Speech API для распознавания ключевых фраз.

Данный выбор сделан из-за того, что для технологии Snowboy не нужны удаленные сервера для распознавания, но для каждого слова требуется создание языковой модели, поэтому данная технология хорошо подходит для распознавания кодового слова, например «Дом». Google Cloud Speech API выбран как раз для того, чтобы распознавать фразы, которые настроят пользователи, без необходимости каждый раз составлять модели.

2.2.8 Вывод

Из проведенного тестирования и последующей обработки результатов был сделан вывод, что SnowBoy наиболее подходит для распознавания ключевого слова, а Google Cloud Speech API для распознавания ключевых фраз.

2.3 Выбор протокола обмена данными

Чтобы осуществить связь между приложением и «Умным домом» необходимо выбрать протокол, с помощью которого мы будем передавать данные.

На сегодняшний день не существует единой платформы и инфраструктуры для поддержки всех устройств домашней автоматизации. На данный момент беспроводные сети представлены множеством технологий, каждая из которых обладает теми или иными преимуществами. Все использующиеся ныне в системах домашней автоматизации протоколы или создавались для иных целей (Wi-Fi для высокоскоростной передачи данных), или же успели безнадежно устареть [20].

Но ни одна из сетей не может полностью удовлетворить все запросы разработчиков и потребителей. У каждой из них присутствует как ряд плюсов, так и минусов. Поэтому все еще отсутствует повсеместное универсальное решение. В результате, наблюдается высокая степень фрагментации рынка IoT [21].

При исследовании упор был поставлен на простоту реализации, обслуживания, распространенность протокола и открытость стандартов.

Разберем наиболее часто применяемые протоколы, их достоинства и недостатки.

2.3.1 MQTT

MQTT (Message Queue Telemetry Transport) – упрощённый сетевой протокол, работающий поверх TCP/IP. Используется для обмена сообщениями

между устройствами по принципу издатель-подписчик. Идеален для использования в контроллерах и датчиках, где требуется небольшой размер кода и есть ограничения по пропускной способности канала [22].

Протокол MQTT работает на прикладном уровне поверх TCP/IP и использует по умолчанию 1883 порт (8883 при подключении через SSL). Также, возможна работа через Winsocket, что позволяется адаптировать его на многие платформы.

Обмен сообщениями в протоколе MQTT осуществляется между клиентом (client), который может быть издателем (publisher), подписчиком (subscriber) или брокером (broker) сообщений [23].

Издатель отправляет данные в брокер, указывая в сообщении определенную тему, топик (topic). Подписчики могут получать разные данные от множества издателей в зависимости от подписки на соответствующие топики. Клиент может быть одновременно и подписчиком, и издателем сообщении.

Преимущества:

- прост в использовании. Протокол представляет собой программный блок без лишней функциональности, который может быть легко встроен в любую сложную систему;

- открытый стандарт;

- шаблон проектирования издатель-подписчик удобен для большинства решений с датчиками. Дает возможность устройствам выходить на связь и публиковать сообщения, которые не были заранее известны или предопределены;

- лёгок в администрировании;

- снижена нагрузка на канал связи;

- работа в условиях постоянной потери связи или других проблем на линии;

- нет ограничений на формат передаваемых данных.

Недостатки:

- из-за не ограничения формата данных возможно несоответствия формата данных у разных производителей оборудования;

- слабая безопасность: в основной не используются методы защиты для передачи данных.

2.3.2 XMPP

XMPP (раньше Jabber) был разработан для системы мгновенного обмена сообщениями для связи между людьми с помощью текстовых сообщений. XMPP означает Extensible Messaging and Presence Protocol, или расширяемый протокол обмена сообщениями и информацией о присутствии. В протоколе XMPP используется текстовый формат XML в качестве встроенного типа, обеспечивая естественную связь между людьми. Протокол работает по TCP/IP. Концепция позволяет использовать данный протокол в IoT [24].

Преимущества:

- децентрализация: Архитектура сети XMPP схожа с электронной почтой; кто угодно может запустить свой собственный XMPP-сервер и нет какого-либо центрального сервера;

- открытый стандарт;

- безопасность: XMPP серверы могут быть изолированы от публичных сетей XMPP (например, во внутренней сети компании) и хорошо защищены.

Недостатки:

- избыточность передаваемой информации: как правило, более 70 % трафика XMPP составляют сообщения о присутствии, около 60 % которых являются излишними, что создает лишнюю нагрузку на канал связи;

- неэффективность передачи бинарных данных: так как XMPP является, по сути, одним длинным XML-документом, невозможно передать не модифицированную двоичную информацию. В результате этого, для передачи файлов стараются использовать дополнительные протоколы, например, HTTP.

2.3.3 AMQP

AMQP (Advanced Message Queuing Protocol) – открытый протокол для передачи сообщений между компонентами системы. Основная идея состоит в том, что отдельные подсистемы (или независимые приложения) могут обмениваться произвольным образом сообщениями через AMQP-брокер, который осуществляет маршрутизацию, возможно гарантирует доставку, распределение потоков данных, подписку на нужные типы сообщений [25].

AMQP основан на трёх понятиях:

Сообщение (message) – единица передаваемых данных, основная его часть (содержание) никак не интерпретируется сервером, к сообщению могут быть присоединены структурированные заголовки.

Точка обмена (exchange) – в неё отправляются сообщения. Точка обмена распределяет сообщения в одну или несколько очередей. При этом в точке обмена сообщения не хранятся. Точки обмена бывают трёх типов:

fanout – сообщение передаётся во все прицепленные к ней очереди;

direct – сообщение передаётся в очередь с именем, совпадающим с ключом маршрутизации (routing key) (ключ маршрутизации указывается при отправке сообщения);

topic – нечто среднее между fanout и direct, сообщение передаётся в очереди, для которых совпадает маска на ключ маршрутизации, например, app.notification.sms.# – в очередь будут доставлены все сообщения, отправленные с ключами, начинающимися с app.notification.sms.

Очередь (queue) – здесь хранятся сообщения до тех пор, пока не будут забраны клиентом. Клиент всегда забирает сообщения из одной или нескольких очередей.

Преимущества:

- открытый стандарт;
- разработан для исключения потерь данных между устройствами;

- ориентирован на банковские системы, что дает предположения о его надежности и безопасности в использовании.

Недостатки:

- сложность в реализации и обслуживании.

2.4 Вывод

Исходя из вышеизложенного было принято решение об использовании протокола MQTT, концепция которого подходит для разработки ПО для взаимодействия с «Умным домом».

3 Реализация приложения

Основными модулями разрабатываемого приложения являются следующие:

- подсистема распознавания речи;
- подсистема пользовательского интерфейса;
- подсистема выполнения команд;
- подсистема хранения данных.

Реализация выполнена на языке Java в среде Android Studio.

Весь код мобильного приложения находится в одной директории «MyHome». Ее структура выглядит следующим образом:

1) Каталог «Application» - содержит исходный код приложения и необходимые для работы приложения файлы:

- каталог «MyHome\app\src\main\java\com\alimovnin\myhome» - содержит исходный код мобильного приложения:

- AppCompatPreferenceActivity.java – вспомогательное описание средств настроек;
- CommandHandler.java – методы выполнения команд;
- Database.java – описания баз данных;
- GoogleRecognize.java – взаимодействие с Google Cloud Speech API;
- MqttClient.java – взаимодействие с API Paho MQTT;
- MyHomeActivity.java – описание главной страницы интерфейса, включающее в себя функции избранного;
- MyHomeService.java – описание службы для взаимодействия с ОС Android, который запускает все приложение;
- HotwordRecognize.java – описание взаимодействия с API SnowBoy;
- SettingsActivity.java – описания интерфейса настроек;
- TopicPreference.java, TopicsActivity.java, TopicsAdapter.java, TopicsList.java – описания интерфейса настроек топика;
- TTSHandler.java – описания взаимодействия с технологией Text-to-speech;

- каталог «MyHome\app\src\main\res» - содержит файлы описания интерфейса и всего текста, выводимого на экран.

2) Каталог «MyHome\app\src\main\java\ai\kitt\snowboy» содержит файлы необходимые для работы SnowBoy.

Файл MyHomeService содержит службу, выполняющую задачи, для которых не нужен графический интерфейс. Служба выделяет место в памяти для служб распознавания, генерации голоса, подключения и взаимодействия с MQTT.

3.1 Реализация голосового интерфейса

Для реализации распознавания команд пользователя был использованы технологии SnowBoy и Google Cloud Speech API [пункт 2.2.7]. Технология SnowBoy работает без подключения к интернету. Google Cloud Speech API может работать как с использованием интернета, так и без него, при условии установленного языкового пакета в ОС Android.

3.1.1 Алгоритм работы голосового интерфейса

Для работы технологии SnowBoy были созданы языковые модели слов «MyHome», «Home», «Дом», «Привет, дом» [пункт 2.2.6].

После подключения к «Умному дому» происходит запуск постоянного распознавания «ключевого слова», которое задано в настройках. Данное распознавание работает до прекращения соединения с MQTT или прекращении работы приложения. Если ключевое слово было определено, то происходит запуск распознавания ключевой фразы. Распознанная информация в текстовом виде передается в подсистему выполнения команд.

3.2 Реализация интерфейса к аппаратной части

Для взаимодействия с «Умным домом» был выбран протокол MQTT [пункт 2.4]. В качестве открытой реализации протокола для ОС Android взят проект Paho

MQTT. Данный проект сочетает в себе весь необходимый функционал для взаимодействия с брокером MQTT [26]. Вся поступающая информация через API сохраняется в базе данных для дальнейшего анализа и использования.

3.2.1 Взаимодействие с протоколом

Для подключения приложения к MQTT брокеру используется метод connect() в которые передаются параметры введенные «Администратором»:

1) Адрес подключения.
2) Имя пользователя.
3) Пароль.
4) Параметры времени задержки передачи и выбранного начального топика.

После подключения приложение (подписчик) начинает считывать с брокера все сообщения вида названия топика и значение, впоследствии чего передает их в базу данных для хранения.

Для отключения вызывается метод disconnect(), где происходит отписка от начального топика и разрыв соединения с брокером.

3.3 Реализация базы данных

Для хранения пользовательских данных и топиков MQTT используется встроенная в Android база данных SQLite 3. Данная база данных имеет функции создания таблиц, создания и изменения существующих данных и использования поиска по необходимым параметрам. В базе данных были созданы 4 таблицы. Каждая таблица подразумевает свое собственное назначение:

1) Для хранения данных MQTT (таблица mqtt).
2) Для хранения кодовых фраз (таблица keyphrase).
3) Для хранения параметров команд (таблица parameters).
4) Для хранения параметров синтеза речи (таблица tts).

Для каждого топика MQTT в базе данных создается запись с определенным id, который сопоставляется с данным топиком во всех таблицах, что позволяет не создавать повторные записи с данным топиком и отсеивать их по нужным параметрам.

Для хранения данных MQTT была создана таблица mqtt, в ней содержится:

1) Название топика.
2) Текущее значение.
3) Дата изменения значения.
4) Назначенная команда.
5) Дополнительные параметры: пользовательское название топика, отображение топика.

Для хранения кодовых фраз была создана таблица keyphrase, в данной таблице содержится заданные пользователями кодовые фразы для распознавания.

В таблице parameters содержатся числовые или символьные значения, на которые будут изменяться значения топиков MQTT после выполнения действий.

Таблица tts предназначена для хранения:

1) Назначенной команды для синтезатора речи.
2) Дополнительный текст при проигрывании синтезируемого текста.
3) Период срабатывания.
4) Дополнительные параметры.

3.3.1 Описание работы базы данных

Для реализации базы данных был создан класс Database, который позволяет создавать базу данных, вызовом конструктора Database() или очищать уже созданную вызовом метода clear().

База данных формируется с помощью метода onCreate, при вызове которого, создаются четыре таблицы – mqtt, keyphrase, parameters, tts.

3.4 Реализация подсистемы выполнения команд

В данной подсистеме происходит поиск распознанной текстовой информации в базе данных, а также выполнение всех необходимых действий со значением топика MQTT. В последствии измененные значения передаются «Умному дому». На рисунке 2 представлена блок-схема подсистемы выполнения команд, также в приложении А приведен код выполнения команд.

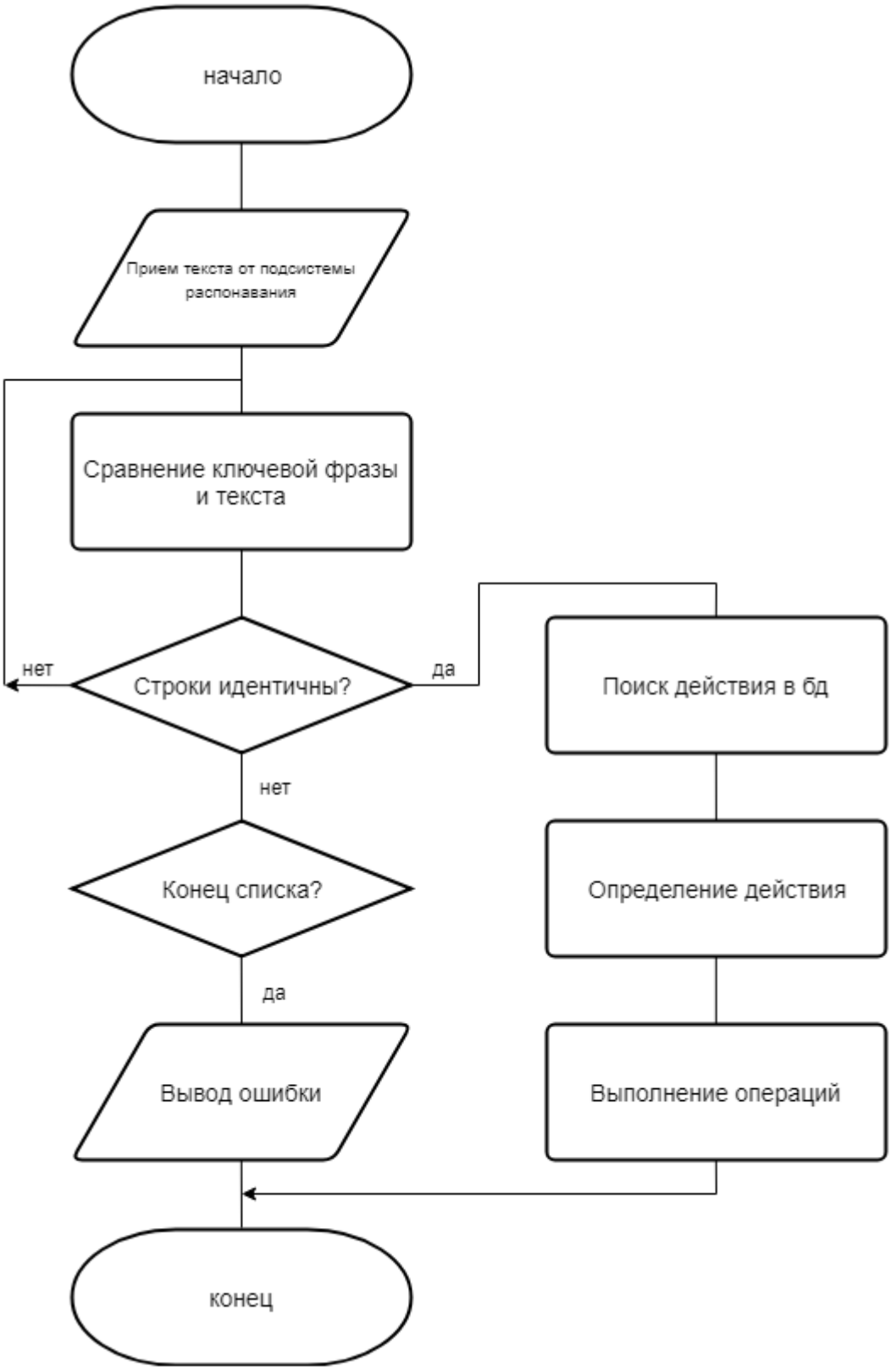

Рисунок 2 – Подсистема выполнения команд

4 Пользовательский интерфейс

Для взаимодействия с приложением был разработан пользовательский интерфейс. Функции интерфейса разделены на две категории:

1) Функции «Администратора».
2) Функции «Клиента».

Мобильное приложение можно условно разделить на три основных экрана пользовательского интерфейса и одну фоновую задачу:

1) Экран «Избранное».
2) Экраны настроек.
3) Уведомление в панели уведомлений ОС Android.
4) Фоновая задача по распознаванию и генерации голоса.

Мобильное приложение можно использовать в двух локализациях языка: «русский» и «английский». Данное решение позволяет расширить пользовательскую аудиторию приложения.

4.1 Интерфейс администратора

В интерфейс для пользователя «Администратор» входит:

- функция настройки приложения;
- функция настройки подключения к MQTT;
- функция настройки топиков.

На рисунке 3 предоставлена диаграмма сценария подключения и настроек топиков.

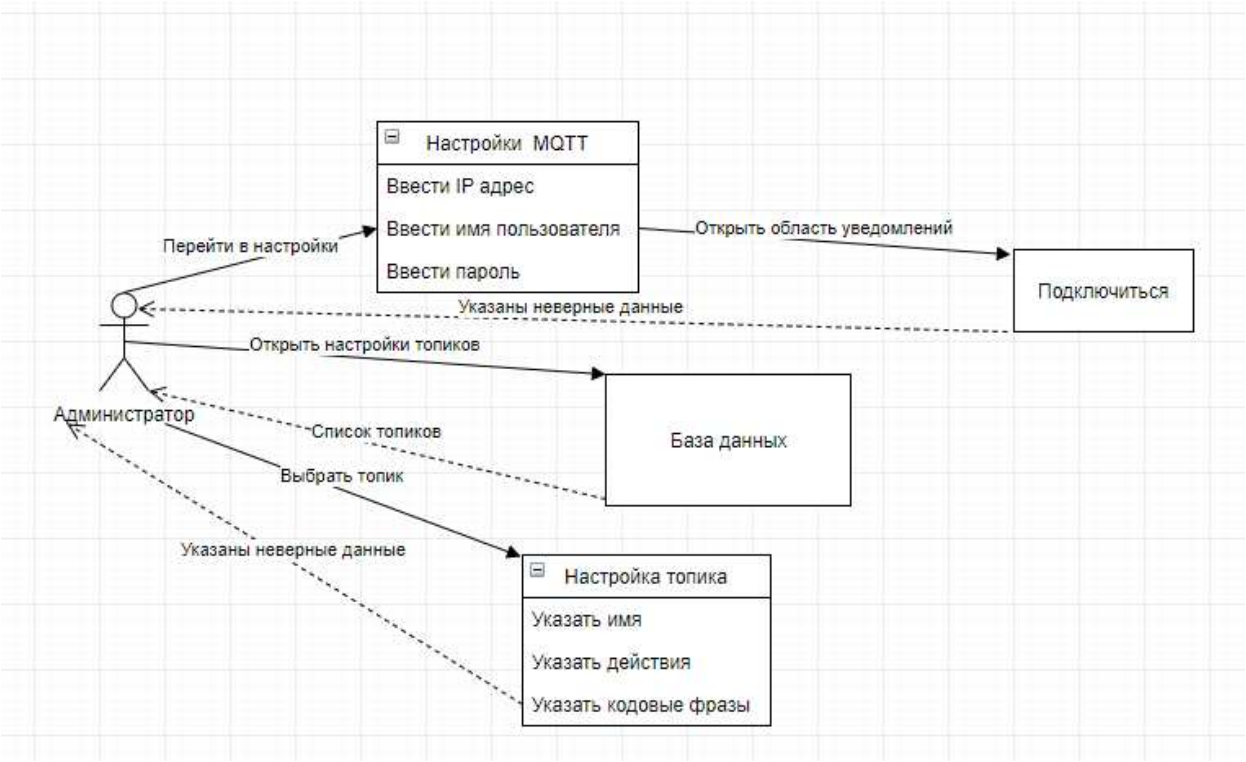

Рисунок 3 – Сценарий использования приложения пользователем «Администратор»

4.1.1 Настройка приложения

Внешний вид основного экрана настроек предоставлен на рисунке 4.

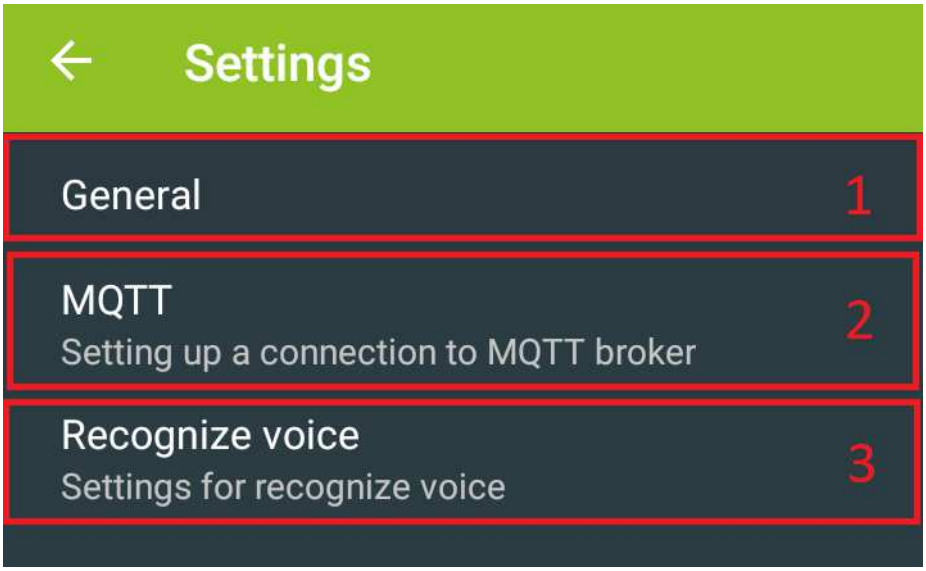

Рисунок 4 – Экран настроек

Экран разделен на три пункта, цифрами обозначены:

1) Настройки приложения.
2) Настройки MQTT.
3) Настройки распознавания и генерации голоса.

4.1.2 Настройки MQTT

Пункт «Настройки MQTT» содержит в себе поля для ввода адреса подключения, начального топика, имени пользователя и пароля. Данные поля необходимо заполнить «Администратору» для дальнейшей корректной работе приложения. Эти настройки показаны на рисунке 5.

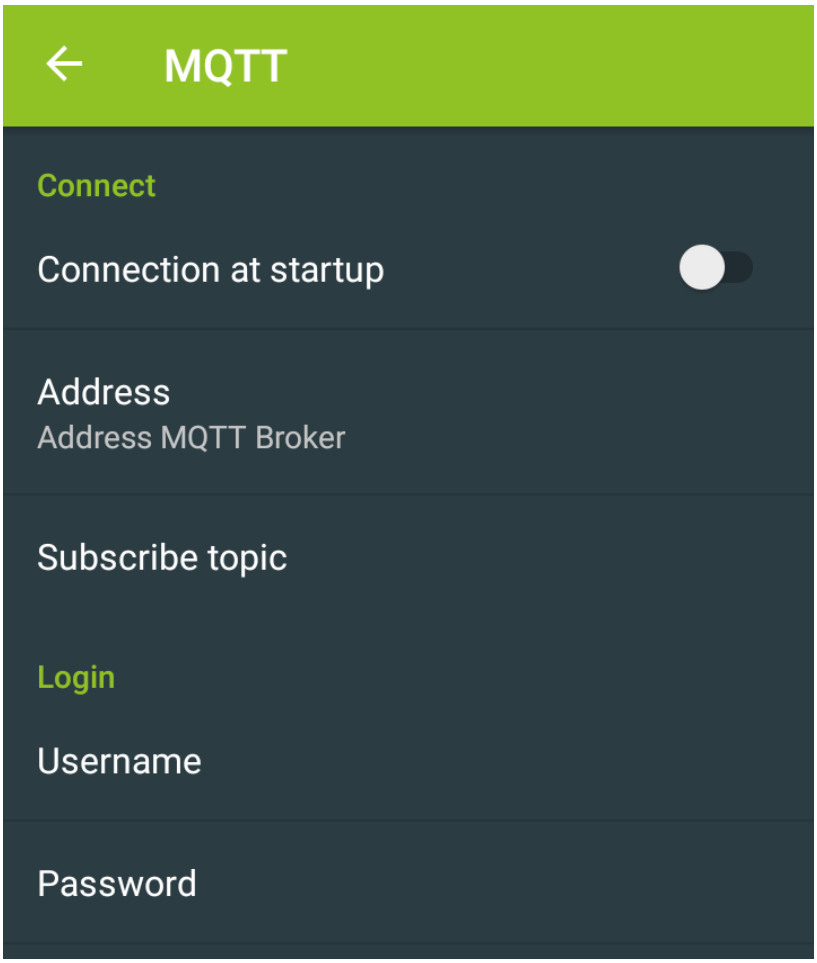

Рисунок 5 – Экран настройки MQTT

4.1.3 Настройки приложения

Пункт «Настройки приложения» содержит в себе три пункта:

1) Список всех топиков.
2) Настройка значений по умолчанию.
3) Сбросить базу данных.

Данные настройки продемонстрированы на рисунке 6.

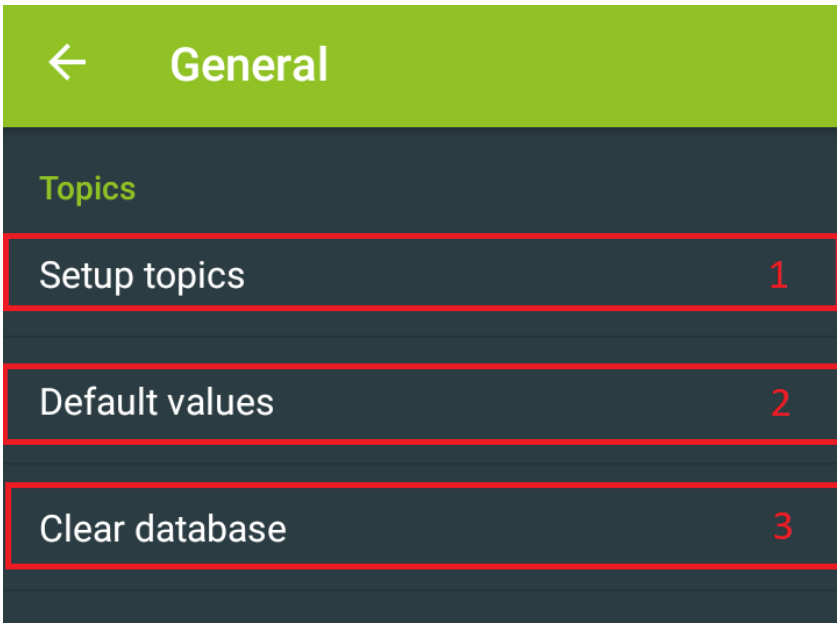

Рисунок 6 – Настройки приложения

4.1.3.1 Настройки топиков

Пункт «Топики», изображенный на рисунке 7, содержит экран всех хранящихся записей в базе данных.

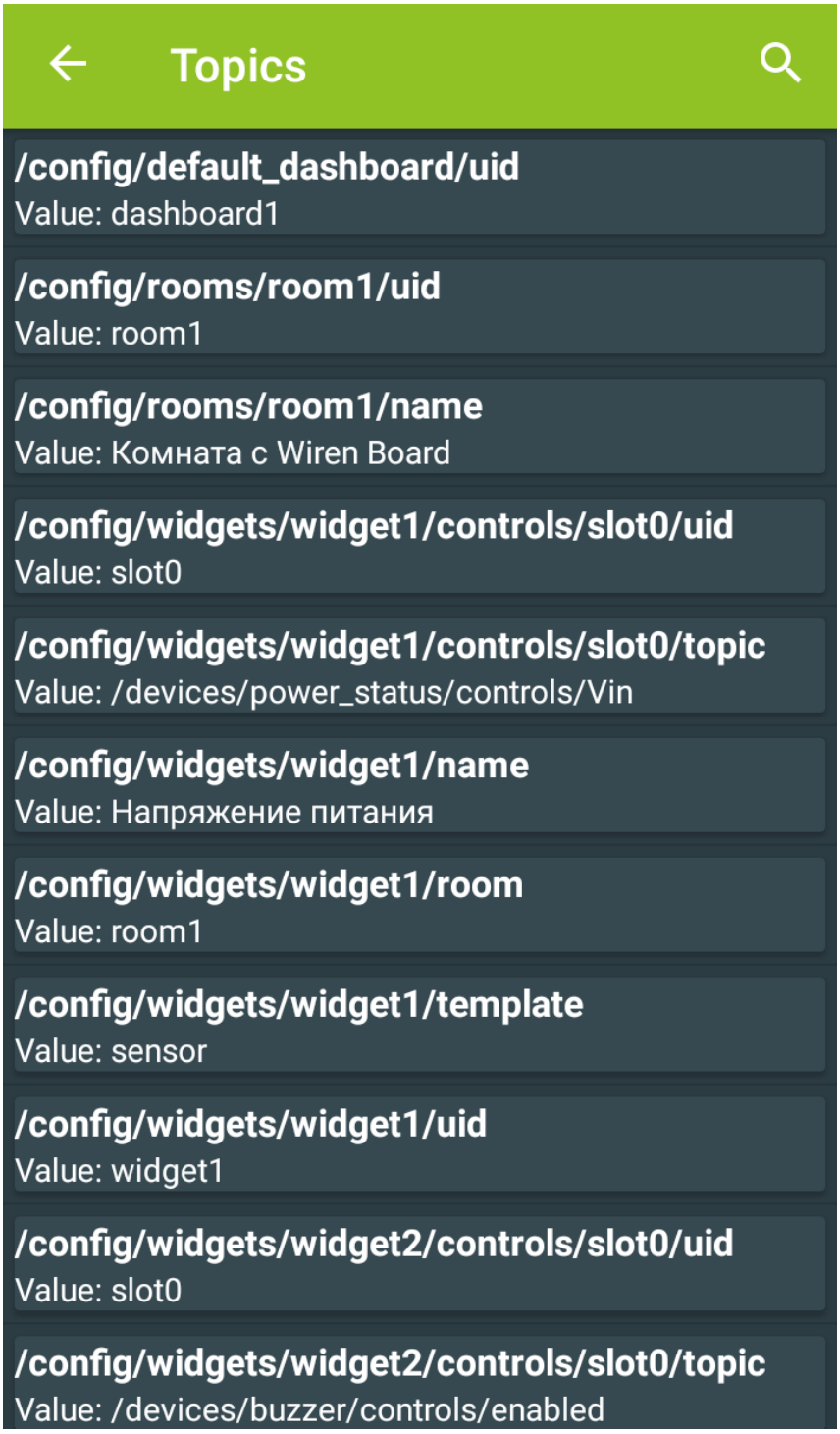

Рисунок 7 – Экран топиков

При выборе любой записи происходит переход к настройкам конкретного топика. Экран настройки топика предоставлен на рисунке 8.

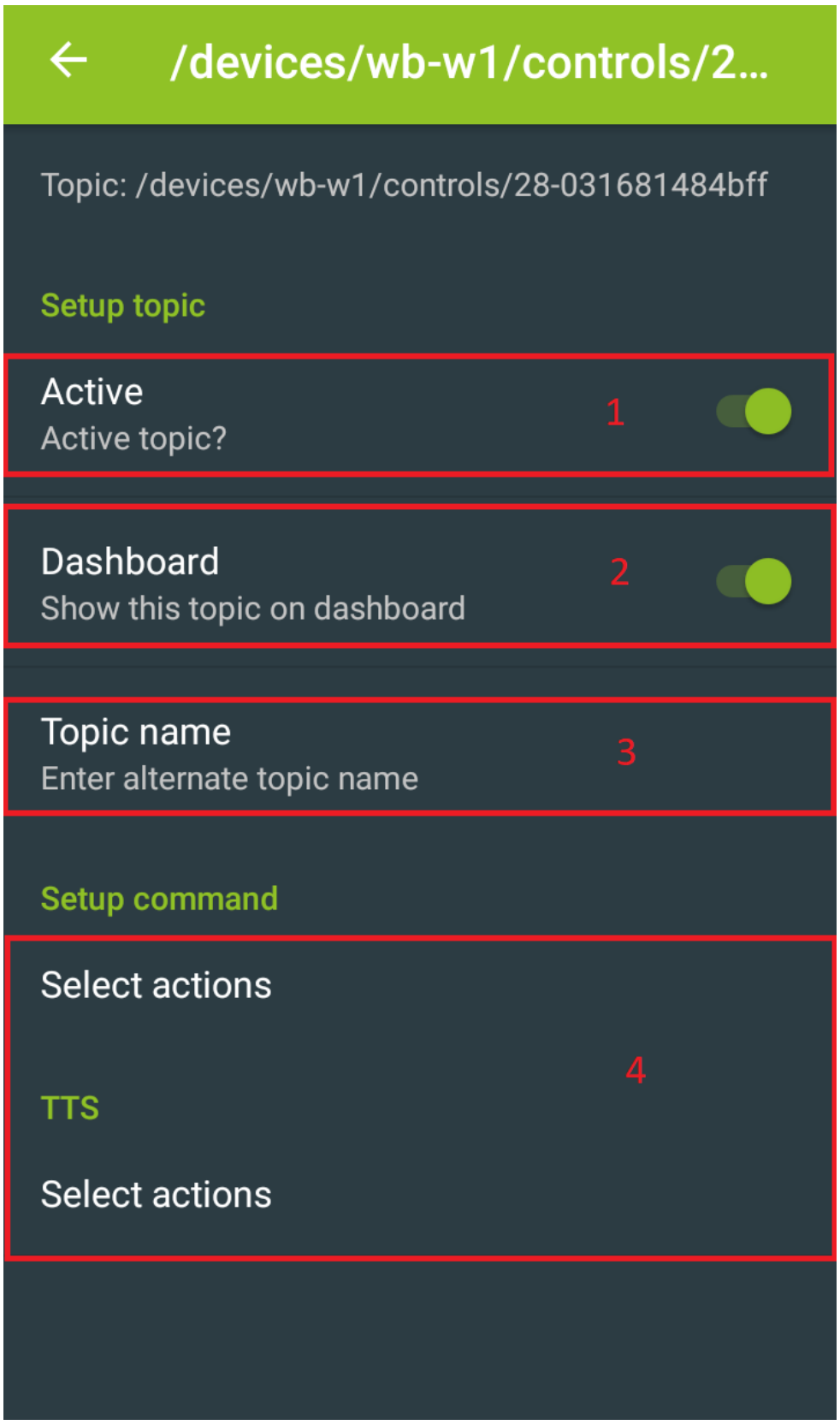

Рисунок 8 – Экран настройки топика

На данном экране пользователь с ролью «Администратор» может настроить:

1) Показывать ли эту запись на экране топиков.
2) Показывать ли эту запись в избранном.
3) Имя топика.
4) Пункты задания команд для распознавания и «действия», которые будут выполнены после распознавания.
5) Пункты и параметры для настроек пользователем «Администратор».
6) Пункт «Выбрать действие» содержит в себе выпадающий список «действий». В приложении заранее включено четыре вида «действий»: «Включить-выключить», «Включить через время», «Включить и выключить через время» и «Увеличить или уменьшить значение», после выбора «действия» появляются дополнительные поля для ввода команд(-ы) и параметров(-а) в зависимости от того, что делает «действие»:

- «включить-выключить» - включает в себя два поля ввода команд для распознавания и два поля для ввода значений, которые будут передаваться по протоколу MQTT. Передаваемые значения могут быть заданы пользователем «Администратор» в пункте настроек «Значения по умолчанию», изменены для конкретного топика в поле ввода значений или могут использоваться стандартные значения, прописанные в программе;

- «включить через время» - включает в себя поле ввода команды, поле для ввода времени и поле для ввода значения;

- «включить и выключить через время» - включает в себя поле ввода команды, поле для ввода времени и два поля ввода значений включения и выключения;

- «увеличить или уменьшить значение» - включает в себя семь полей для ввода. Четыре поля предназначены для указания команд: «Увеличить», «Уменьшить», «Установить максимальное значение», «Установить минимальное значение». Три поля отведено для установки параметров максимального,

минимального значения, и величины, на которую нужно изменить значение топика.

7) Пункт «Выбрать действие для TTS» содержит в себе выпадающий список «действий» для генерации голоса и включает в себя два «действия»:

- «сказать значение один раз» - включает в себя поле ввода команды и поле ввода текста для синтеза речи. Данный текст и значение из топика будет сказано после распознавания заданной команды;

- «сказать значение через…» - включает в себя четыре поля ввода. Два поля отведено для команд «Запустить таймер» и «Остановить таймер», поле ввода текста для синтеза речи, и поле ввода времени. Данный текст и значение из топика будет сказано с определенной периодичностью. Запуск и остановка происходит после распознавания команды.

4.1.3.2 Настройка значений по умолчанию

Для изменения значений по умолчанию используется пункт «Значения по умолчанию» изображенный на рисунке 9.

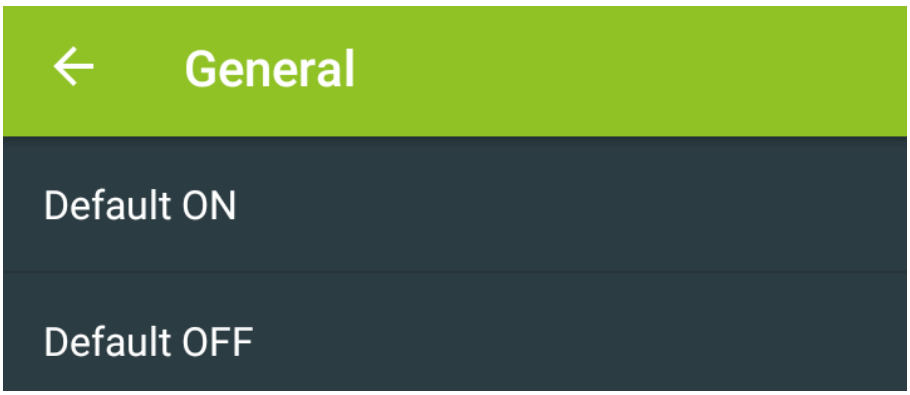

Рисунок 9 – Настройка значений по умолчанию

4.2 Интерфейс клиента

На рисунке 10 предоставлена диаграмма сценария подключения и использовании функции голосового управления.

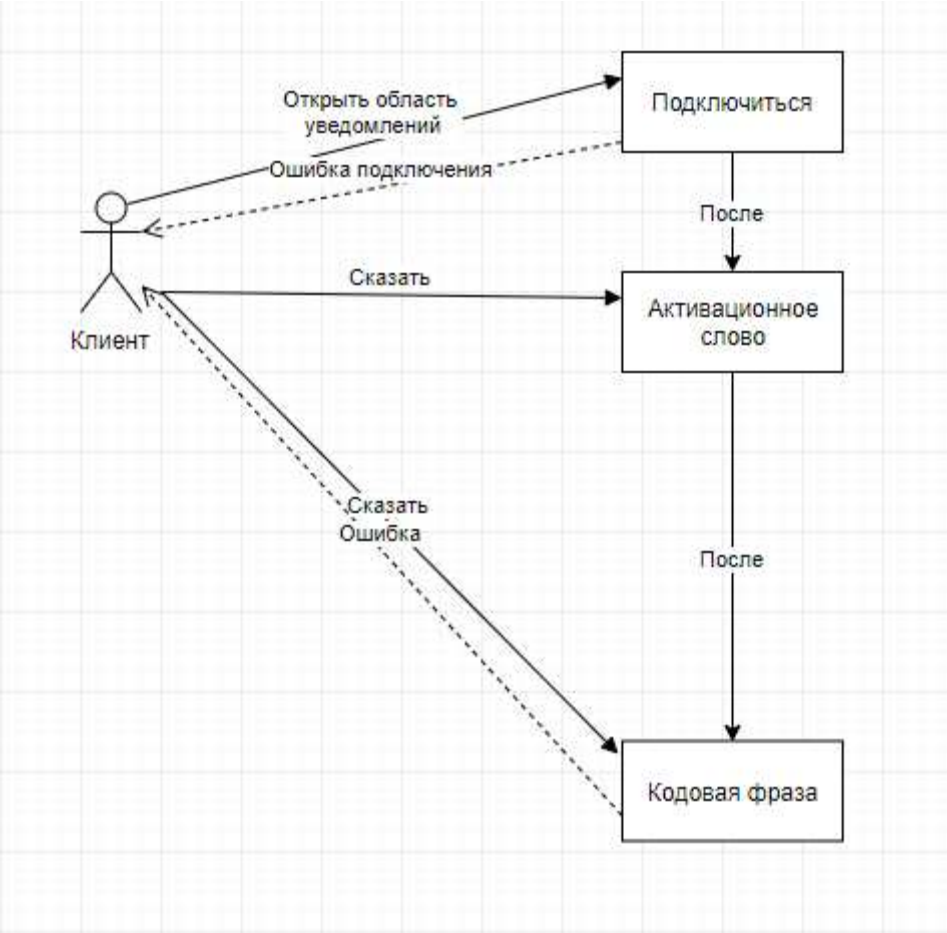

Рисунок 10 – Сценарий пользователя «Клиент»

В интерфейс для пользователя «Клиент» входит:

- функция настройки избранных топиков;

- функция подключения к MQTT, при условии, что пользователь с ролью «Администратор» настроил до этого подключение;

- функция дальнейшего использования: произношение фразы и получение обратной связи в виде синтеза речи или всплывающего сообщения.

4.2.1 Избранное

Внешний вид избранного предоставлен на рисунке 11. Цифрами обозначены:

8) Кнопка вызова меню

9) Список избранного

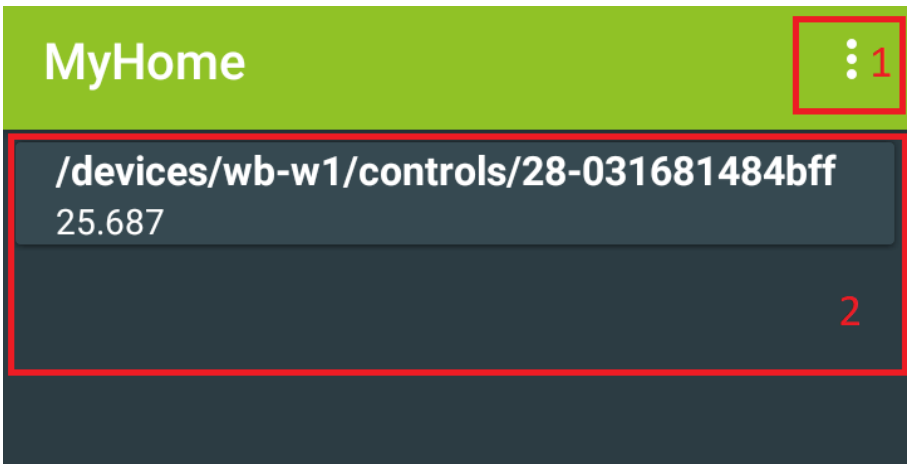

Рисунок 11 – Главный экран приложения

Список избранного включает в себя список часто просматриваемых для «Администратора» и «Пользователя» топиков MQTT. Каждая запись состоит из назначенного пользователями названия топика и его текущего значения.

Меню избранного содержит два пункта: «Настройки» и «О приложении». Кнопка «Настройки» позволяет перейти в настройки приложения и MQTT. Всплывающее окно «О приложении» содержит в себе информацию о разработчике приложения и краткую помощь в первоначальной настройке. Данное окно предоставлено на рисунке 12.

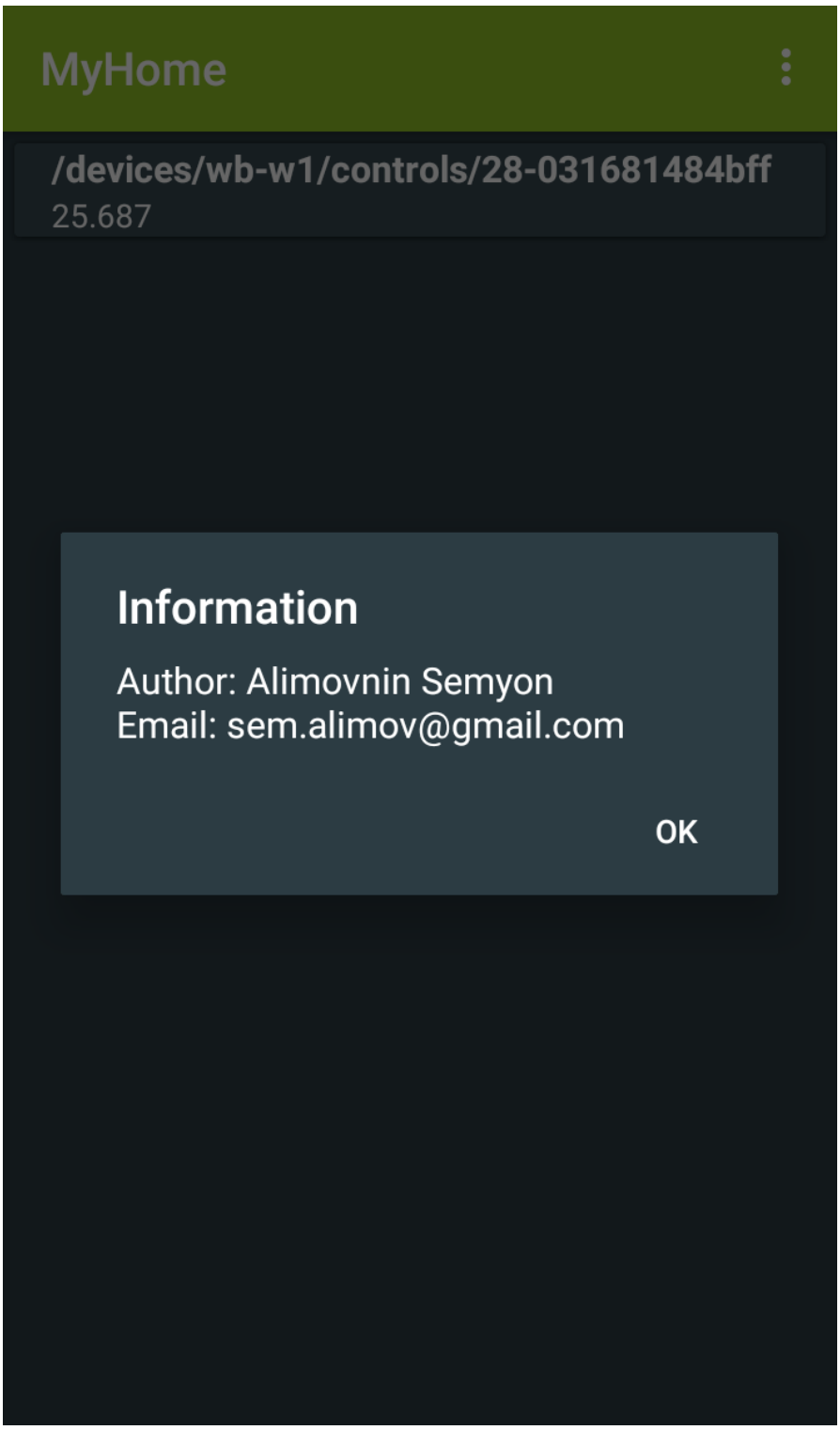

Рисунок 12 – О приложении

4.2.2 Интерфейс подключения

Интерфейс для подключения представляет собой уведомление в области уведомлений ОС Android.

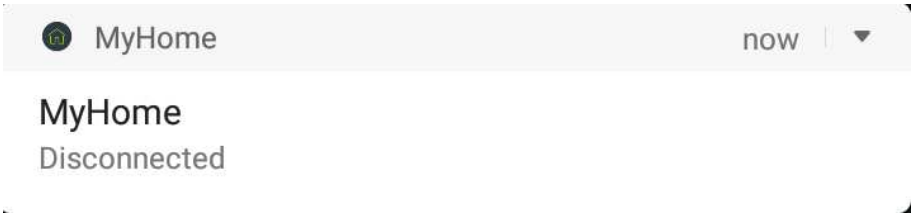

Рисунок 13 – Уведомление при отсутствии соединения с MQTT

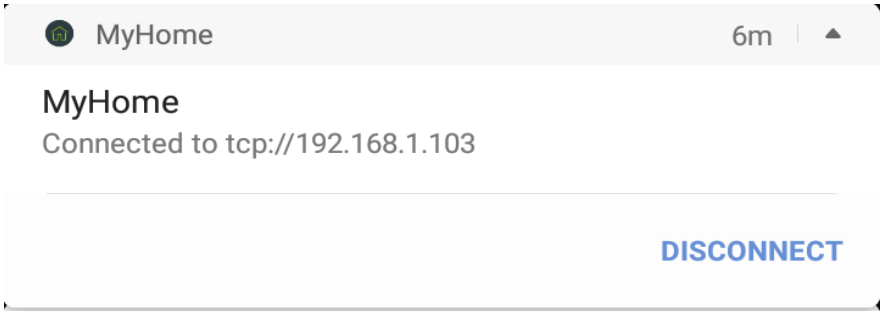

Рисунок 14 – Уведомление при подключении к MQTT

При нажатии кнопки «Присоединиться» или «Отсоединиться», в зависимости от статуса подключения, происходит изменение уведомления.

4.3 Тестирование приложения

Для тестирования приложения были проведены тесты в условиях существующей системы «Умный дом». Пользователем "Администратор» были проведены следующие тесты.

Настройка подключения, с вводом корректного и ошибочного адреса подключения. В результате ввода ошибочного адреса подключение не было совершено и вывелась ошибка подключения. Для корректного адреса соединение было успешно установлено.

Настройка топиков MQTT с вводом одинаковых ключевых фраз, в результате программа сообщила об ошибке, что невозможно использовать ключевую фразу дважды.

Для пользователя «Клиент» был проведен следующие тесты.

Настройка двух топиков, отвечающих за температуру и включение или выключения света.

Произнесение ключевого слова «Дом» и ключевой фразы для каждого теста. Для топика, отвечающего за температуру внутри помещения, было выбрано действие «Сказать значение один раз» и назначена фраза «Скажи температуру». Для топика, отвечающего за свет, было выбрано действие «Включить-выключить» и назначена фраза «Включи».

В ходе тестирования приложения получены следующие результаты, что приложение не может правильно определить кодовую фразу при очень тихой речи. Также были замечены случайные срабатывания ключевого слова «Дом», но из-за того, что необходимо еще добавить ключевую фразу критического влияния на систему «Умный дом» этого не оказывает. Для того, чтобы избежать случайного срабатывания системы было принято решение в дальнейшем добавить функцию подтверждения потенциально опасных команд.

ЗАКЛЮЧЕНИЕ

В рамках решения задач выпускной квалификационной работы были получены следующие результаты:

Был проведен анализ существующих аналогов интерфейса голосового управления. Для анализа были выбраны три технологии – Google Home, Amazon Echo и Apple Siri. В результате был сделан вывод, что данные технологии являются дорогостоящими для интеграции в системы «Умный дом»;

Проведен сбор и анализ информации о существующих технологиях голосового распознавания и взаимодействия с «Умным домом». В результате анализа и тестирования технологий был сделан вывод, что Google Cloud Speech API наиболее подходит для распознавания команд, а SnowBoy для распознавания ключевого слова.

Спроектировано и реализовано мобильное приложение, включающее в себе функционал голосового управления системой «Умный дом».

Приложение было протестировано и показало свою работоспособность в реальных условиях системы «Умный дом», что позволяет использовать реализованное приложение в дальнейшем для пользователей.

В процессе работы были опубликованы две статьи в сборниках материалов конференций, получены два диплома первой степени [27, 28].

В дальнейших планах выпустить данное приложение в Google Play для свободного скачивания и использования, планируется развитие и добавление нового функционала.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Языки для «Умного дома» [Электронный ресурс] // Бизнес новости. – Режим доступа: http://www.forbes.ru

2 Что такое голосовые технологии [Электронный ресурс] // Теплица социальных технологии. – Режим доступа: http://te-st.ru

3 Голосовой интерфейс [Электронный ресурс] // Википедия. – Режим доступа: http://ru.wikipedia.org

4 Экосистема Amazon Alexa [Электронный ресурс] // Geektimes. – Режим доступа: http://geektimes.ru

5 Google Home [Электронный ресурс] // Википедия. – Режим доступа: http://ru.wikipedia.org

6 Почему «умные» голосовые помощники пока никому не помогают [Электронный ресурс] // Главные новости. – Режим доступа: https://www.gazeta.ru

7 Siri [Электронный ресурс] // Национальная библиотека им Н.Э. Баумана. – Режим доступа: https://ru.bmstu.wiki

8 HomeKit [Электронный ресурс] // Apple. – Режим доступа: https://www.apple.com

9 Технологии распознавания речи [Электронный ресурс] // ПостНаука. – Режим доступа: https://postnauka.ru

10 Сверхбыстрое распознавание речи без серверов на реальном примере [Электронный ресурс] // Habrahabr. – Режим доступа: https://habrahabr.ru

11 Они нас слышат: куда развиваются речевые технологии? [Электронный ресурс] // Бизнес новости. – Режим доступа: http://www.forbes.ru

12 Распознавание речи для чайников [Электронный ресурс] // Habrahabr. – Режим доступа: https://habrahabr.ru

13 Комплекс речевых технологий Яндекса [Электронный ресурс] // Яндекс. – Режим доступа: https://tech.yandex.ru

14 Cloud Speech-To-Text [Электронный ресурс] // Google Cloud. – Режим доступа: https://cloud.google.com

15 Поиск оптимальной системы аудио распознавания речи с закрытым исходным кодом, но имеющими открытые API, для возможности интеграции [Электронный ресурс] // Habrahabr. – Режим доступа: https://habrahabr.ru

16 Google обновила Cloud Speech API, добавив поддержку 30 языков [Электронный ресурс] // Создано программистами для программистов. – Режим доступа: https://tproger.ru

17 Google создал систему распознавания речи без подключения к интернету [Электронный ресурс] // Интернет издание о высоких технологиях. – Режим доступа: http://www.cnews.ru

18 Speak to Alexa on Products with Alexa Voice Service Built-In [Электронный ресурс] // Alexa Voice Service. – Режим доступа: https://developer.amazon.com

19 Hotword Detector [Электронный ресурс] // Snowboy. – Режим доступа: https://snowboy.kitt.ai

20 Умный дом: Развитие и тенденции [Электронный ресурс] // Geektimes. – Режим доступа: https://geektimes.ru

21 Протоколы связи для "умного дома" [Электронный ресурс] // Аналитические обзоры компьютеров. – Режим доступа: https://www.ferra.ru

22 MQTT протокол теперь в устройствах Advantech [Электронный ресурс] // Комплексная дистрибуция средств автоматизации. – Режим доступа: http://corson.ru

23 Что такое MQTT и для чего он нужен в IIoT? [Электронный ресурс] // Промышленные компьютеры. – Режим доступа: https://ipc2u.ru

24 Протоколы «Интернета вещей»: основные сведения [Электронный ресурс] // Средства и системы автоматизации. – Режим доступа: http://www.rtsoft.ru

25 RabbitMQ: Введение в AMQP [Электронный ресурс] // Habrahabr. – Режим доступа: https://habrahabr.ru

26 MQTT and MQTT-SN messaging protocols [Электронный ресурс] // Eclipse Paho. – Режим доступа: https://www.eclipse.org

27 Специальные возможности системы "Умный дом". Электронный сборник статей победителей V Международной научно-практической конференции «EUROPEAN SCIENTIFIC CONFERENCE» 30 июля 2017г. в г. Пенза. Электронный сборник МК-196 [Электронный ресурс]/Алимовнин С.Г., Мильчаков С.А., Сиротинина Н.Ю. // Наука и Просвещение. – Пенза, 2017. – Режим доступа: http://naukaip.ru

28 Реализация речевого интерфейса системы «Умный дом» “Профессионал года 2017" Сборник статей V Международного научно-практического конкурса, состоявшейся 25 августа 2017 г. в г. Пенза. Номер сборника К-56 страницы 52-56 [Электронный ресурс]/Алимовнин С.Г., Мильчаков С.А., Сиротинина Н.Ю. // Наука и Просвещение. – Режим доступа: http://naukaip.ru

ПРИЛОЖЕНИЕ А

Класс обработки команд

```
class CommandHandler {
      //инициализация переменных
      private ContentValues mContentValues;
      private Cursor mCursor;
      private MQTTClient mMQTTClient;

      //конструктор
      CommandHandler(MQTTClient MQTTClient) {
            mContentValues = new ContentValues();
            mMQTTClient = MQTTClient;
      }

      void Handler(String result) {//метод принимает распознанный текст
            //поиск текста в базе данных
            mCursor = mSQLiteDatabase.query(true, Database.TABLE_NAME_KEYPHRASE,
            new String[]{Database.id, Database.phrase_1, Database.phrase_2,
                  Database.phrase_3, Database.phrase_4, Database.TTSPhrase_1,
                  Database.TTSPhrase_2},
                  "UPPER(" + Database.phrase_1 + ") LIKE '%" + result + "%' OR " +
                  "UPPER(" + Database.phrase_2 + ") LIKE '%" + result + "%' OR " +
                  "UPPER(" + Database.phrase_3 + ") LIKE '%" + result + "%' OR " +
                  "UPPER(" + Database.phrase_4 + ") LIKE '%" + result + "%' OR " +
                  "UPPER(" + Database.TTSPhrase_1 + ") LIKE '%" + result + "%' OR
            " +
                  "UPPER(" + Database.TTSPhrase_2 + ") LIKE '%" + result + "%'",
            null, null, null, null, null);
            if (mCursor.getCount() > 0) {//если было совпадение
                  mCursor.moveToFirst();
                  if (Objects.equals(result,
                  mCursor.getString(mCursor.getColumnIndexOrThrow(Database.phrase_
                  1)))) {
                        Command(mCursor.getString(mCursor.getColumnIndexOrThrow(Dat
                        abase.id)), Database.phrase_1);
                  } else if (Objects.equals(result,
                  mCursor.getString(mCursor.getColumnIndexOrThrow(Database.phrase_
                  2)))) {
                        Command(mCursor.getString(mCursor.getColumnIndexOrThrow(Dat
                        abase.id)), Database.phrase_2);
                  } else if (Objects.equals(result,
                  mCursor.getString(mCursor.getColumnIndexOrThrow(Database.phrase_
                  3)))) {
                        Command(mCursor.getString(mCursor.getColumnIndexOrThrow(Dat
                        abase.id)), Database.phrase_3);
                  } else if (Objects.equals(result,
                  mCursor.getString(mCursor.getColumnIndexOrThrow(Database.phrase_
                  4)))) {
                        Command(mCursor.getString(mCursor.getColumnIndexOrThrow(Dat
                        abase.id)), Database.phrase_4);
                  } else if (Objects.equals(result,
                  mCursor.getString(mCursor.getColumnIndexOrThrow(Database.TTSPhra
                  se_1)))) {
                        Command(mCursor.getString(mCursor.getColumnIndexOrThrow(Dat
                        abase.id)), Database.TTSPhrase_1);
```

```
            } else if (Objects.equals(result,
            mCursor.getString(mCursor.getColumnIndexOrThrow(Database.TTSPhra
            se_2)))) {
                  Command(mCursor.getString(mCursor.getColumnIndexOrThrow(Dat
                  abase.id)), Database.TTSPhrase_2);
            }
      }
}
//метод обработки действия
private void Command(String id, String column) {
      mCursor.close();
      mCursor = mSQLiteDatabase.query(true, Database.TABLE_NAME_MQTT,
      new String[]{Database.id, Database.command, Database.value,
      Database.topic},
      Database.id + "=?", new String[]{id}, null, null, null, null);
      mCursor.moveToFirst();
      String value =
      mCursor.getString(mCursor.getColumnIndexOrThrow(Database.value));
      final String topic =
      mCursor.getString(mCursor.getColumnIndexOrThrow(Database.topic));
      //если текст был найден в таблице кодовых фраз
      if(mCursor.getString(mCursor.getColumnIndexOrThrow(Database.command))!
      =null) {
            //сравниваем тексты действий
            switch
            (mCursor.getString(mCursor.getColumnIndexOrThrow(Database.comman
            d))) {
                  //если это действие «Включить-выключить»
                  case "activate_deactivate":
                        mCursor.close();
                        //берем из бд текущее значении топика MQTT
                        mCursor = mSQLiteDatabase.query(true,
                        Database.TABLE_NAME_PARAMETERS,
                        new String[]{Database.id, Database.parameters_1,
                        Database.parameters_2},
                        Database.id + "=?", new String[]{id}, null, null,
                        null, null);
                        //если действие «Включить»
                        if (Objects.equals(column, Database.phrase_1)) {
                              value =
                              mCursor.getString(mCursor.getColumnIndexOrThrow
                              (Database.parameters_1));
                        //если действие «Выключить»
                        } else if (Objects.equals(column, Database.phrase_2))
                        {
                              value =
                              mCursor.getString(mCursor.getColumnIndexOrThrow
                              (Database.parameters_2));
                        }
                        try {
                              //пытаемся опубликовать данные в брокер MQTT
                              mMQTTClient.publishMessage(topic, value);
                        } catch (IllegalArgumentException e) {
                              //иначе получаем ошибку
                              e.printStackTrace();
                        }
                        break;
            }
      //если текст был найден в таблице синтеза речи
      } else {
            mCursor.close();
```

```
                    //берем из бд текущее значении топика MQTT
                    mCursor = mSQLiteDatabase.query(true, Database.TABLE_NAME_TTS,
                          new String[]{Database.id, Database.TTS, Database.TTSText},
                    Database.id + "=?", new String[]{id}, null, null, null, null);
                    mCursor.moveToFirst();
                    if(mCursor.getString(mCursor.getColumnIndexOrThrow(Database.TTS)
                    )!=null) {
                          switch(mCursor.getString(mCursor.getColumnIndexOrThrow(Dat
                          abase.TTS))) {

                                case "value_on_command":
                                      String text =
                                      mCursor.getString(mCursor.getColumnIndexOrThrow
                                      (Database.TTSText));
                                      if(text!=null){
                                            MyHomeService.mTTSHandler.speak(text + "
                                            " + value);
                                      } else {
                                          MyHomeService.mTTSHandler.speak(value);
                                      }
                                      break;
                          }
                    }
              }
        }
}
```

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт
Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О.В.Непомнящий
подпись инициалы, фамилия
« ___» _____ 2018 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Голосовое управление системой «Умный дом»
тема

Руководитель	_____ подпись, дата	доцент, канд. техн. наук должность, ученая степень	Н.Ю. Сиротинина инициалы, фамилия
Выпускник	_____ подпись, дата		С.Г. Алимовнин инициалы, фамилия
Нормоконтролер	_____ подпись, дата	доцент, канд. техн. наук должность, ученая степень	В.И. Иванов инициалы, фамилия

Красноярск 2018