

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт Математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем

(код и наименование направления подготовки, специальности)

Web-дизайн и мультимедиа

(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка программного обеспечения для автономного распознавания речи»

Обучающийся

М.А. Шишов

(Инициалы Фамилия)

(личная подпись)

Руководитель

к.т.н., В.С. Климов

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

к.ф.н., М.М. Бажутина

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Аннотация

Тема бакалаврской работы – «Разработка программного обеспечения для автономного распознавания речи».

В настоящее время технологии распознавания речи используются при решении задач человеко-машинного взаимодействия. Например, с помощью голосовых команд можно управлять устройствами в системе «Умный дом». Процесс распознавания речи выполняется на облачных серверах крупных IT компаний (Yandex, Google, Amazon). Так как существуют риски ограничения деятельности этих компаний на территории России, то актуален вопрос создания автономных систем распознавания речи, не требующих для своей работы доступа к сети Интернет. Бакалаврская работа посвящена созданию автономной системы распознавания речи.

Цель работы – разработка программного обеспечения для автономного распознавания речи.

Объектом исследования является технология распознавания речи.

Предмет исследования – автономная реализация системы распознавания речи.

Практическая значимость работы заключается в разработке программного обеспечения, реализующего технологию автономного распознавания речи.

Во введении описывается актуальность, а также цели и задачи исследования. В первой главе бакалаврской работы рассматриваются существующие технологии распознавания речи. Во второй главе приводится архитектура системы для автономного распознавания речи. В третьей главе работы описывается разработанное мобильное приложение. В заключении приводятся результаты бакалаврской работы.

Выпускная квалификационная работа состоит из 40 страниц, 16 рисунков, 1 таблицы и 20 литературных источников.

Abstract

The topic of the bachelor's thesis is "Software Development for Autonomous Speech Recognition".

Speech recognition technologies are currently used in human-machine interaction tasks. For example, voice commands can be used to control devices in a Smart Home system. The speech recognition process is performed on the cloud servers of large IT companies (Yandex, Google, Amazon). Since there are risks of restricting the activities of these companies in Russia, the question of creating autonomous speech recognition systems that do not require Internet access for their work is relevant. This bachelor's thesis is devoted to the creation of an autonomous speech recognition system.

The aim of the work is to develop software for autonomous speech recognition.

The object of the study is speech recognition technology.

The subject of research is the autonomous implementation of a speech recognition system.

The practical significance of the work is the development of software that implements the technology of autonomous speech recognition.

The introduction describes the relevance, as well as the aims and objectives of the research. The first chapter of the bachelor's thesis deals with the existing speech recognition technologies. The second chapter presents the architecture of the system for autonomous speech recognition. The third chapter of the paper describes the developed mobile application. In the conclusion the results of the bachelor's thesis are presented.

The graduation thesis consists of 40 pages, 16 figures, 1 table and 20 references.

Оглавление

Введение.....	5
Глава 1 Анализ технологии распознавания речи.....	8
1.1 Алгоритм распознавания речи.....	8
1.2 Классификация рисков использования облачных сервисов для распознавания речи.....	11
Глава 2 Проектирование системы автономного распознавания речи	14
2.1 Функции системы распознавания речи	14
2.2 Серверная часть системы распознавания речи.....	16
2.3 Выбор протокола для обмена данными.....	18
Глава 3 Разработка мобильного приложения для распознавания речи.....	22
3.1 Проектные решения, применяемые в мобильном приложении	22
3.2 Описание пользовательского интерфейса.....	25
Заключение	36
Список используемой литературы и используемых источников.....	38

Введение

Технологии распознавания речи используются в таких областях, как речевое управление устройствами, организация голосового ввода текста, голосовой поиск, организация речевого интерфейса при человеко-машинном взаимодействии [2], [5], [7].

Наиболее известными примерами применения технологий распознавания для решения практических задач являются:

- создание телефонных роботов для горячих линий, которые позволяют компании экономить человеческие ресурсы компаний при решении наиболее распространенных запросов клиентов;
- речевое управление элементами системы «Умный дом», например, управление светом, электронными замками, системой отопления и т.д. [4];
- разработка речевых интерфейсов для бытовой техники, например, голосовое задание режима стрики, управление телевизором [9];
- голосовые команды в различных десктопных приложениях, в том числе в компьютерных играх [11];
- речевое управление бортовыми системами автомобиля, в том числе – задание адреса конечной точки маршрута в навигационной системе, управление громкостью звука магнитолы и т.д. [14]

Рост популярности задач, требующих применение технологий распознавания речи, привел к созданию облачных сервисов с программным интерфейсом, предоставляющих выполнение данной услуги. Наиболее известными облачными сервисами являются – Yandex.SpeechKit (сервис от компании Яндекс), Google Cloud Speech API (сервис от компании Google) и Alexa Voice Service (сервис от компании Amazon) [19], [22].

Использование облачных сервисов распознавания речи при разработке программного обеспечения обладает следующими достоинствами [17]:

- высокая точность распознавания речи, которая обусловлена постоянно пополняющейся обучающей выборкой речевых данных;
- простота внедрения функций распознавания речи за счет программных интерфейсов (API) предоставляемых сервисами;
- отсутствие необходимости разбираться в алгоритмах искусственного интеллекта применяемых в распознавании речи.

Однако применение облачных сервисов в программных проектах сопряжено с рядом рисков. Например, один из рисков связан с изменением тарифов по оплате услуг облачного сервиса, так тарифы устанавливаются собственниками сервисов в одностороннем порядке.

Помимо экономических рисков, существуют и политические риски, при которых зарубежные компании откажутся предоставлять услуги на основе географического расположения своих клиентов. В этом случае программное обеспечение, в которое интегрированы услуги облачного сервиса, перестанет функционировать.

Также стоит отметить, что у большинства IT компаний бизнес не ограничен одним сервисом, например помимо сервиса распознавания речи компания Google владеет видеохостингом, системой информационного поиска, социальной сетью и пр. Нарушение российских законов по одному направлению может в итоге привести к блокированию всех или большей части сервисов, в том числе и облачного сервиса распознавания речи.

Как становится ясно, реализация любого из описанных рисков может сделать невозможным использование облачного сервиса распознавания речи, что приведет к неработоспособности связанного с ним программного обеспечения. Поэтому актуальным вопросом является проведение исследований и разработка технологий автономного распознавания речи. Автономность заключается в возможности выполнения задач по анализу голоса без использования сети Интернет.

Таким образом актуальной является цель исследования – разработка программного обеспечения для автономного распознавания речи.

Для достижения поставленной цели в работе решаются следующие задачи:

- проведение анализа существующих технологий распознавания речи и применяемых алгоритмов;
- проектирование системы автономного распознавания речи;
- разработка мобильного приложения, реализующего предложенные подходы автономного распознавания речи.

Объектом исследования является технология распознавания речи. Предмет исследования – автономная реализация системы распознавания речи.

Практическая значимость работы заключается в разработке программного обеспечения, реализующего технологию автономного распознавания речи.

Глава 1 Анализ технологии распознавания речи

1.1 Алгоритм распознавания речи

Алгоритм распознавание речи с использованием облачного сервиса Yandex.SpeechKit включает в себя следующие этапы (рисунок 1) [1]:

- запись звука, содержащего речь, посредством встроенного микрофона на используемом устройстве (телефон, телевизор, станция голосового управления и т.д.);
- передача записанного звука в на сервер Yandex, котором функционирует обученная модель распознавания;
- перед запуском процесса распознавания, звуковые данные подвергаются предобработке путем разделения их на фреймы длиной 25 мс;
- следующим этапом является векторизации каждого фрейма, т.е. перевод фрейма в вектор, состоящий из 40 числовых значений;
- полученный вектор передается в модель для распознавания, которая на выходе генерирует вероятность относительно того, какой букве относится данный фрейм;
- результаты распознавания фреймов объединяются в слова, которые корректируются с помощью языковой модели.

Под акустической моделью понимается классификатор, который на основе признаков фрагмента звуковых данных определяет произносимый пользователем звук [3]. В роли классификатора могут использоваться стандартные модели из области машинного обучения – многослойные нейронные сети, набор деревьев решений и т.д. [4].

В простейшем случае под языковой моделью понимается словарь слов и словосочетаний с частой использования элементов словаря. Частота использования элементов словаря применяется, когда не удается достоверно распознать речь пользователя [6].



Рисунок 1 – Алгоритм распознавания речи на примере Yandex.SpeechKit

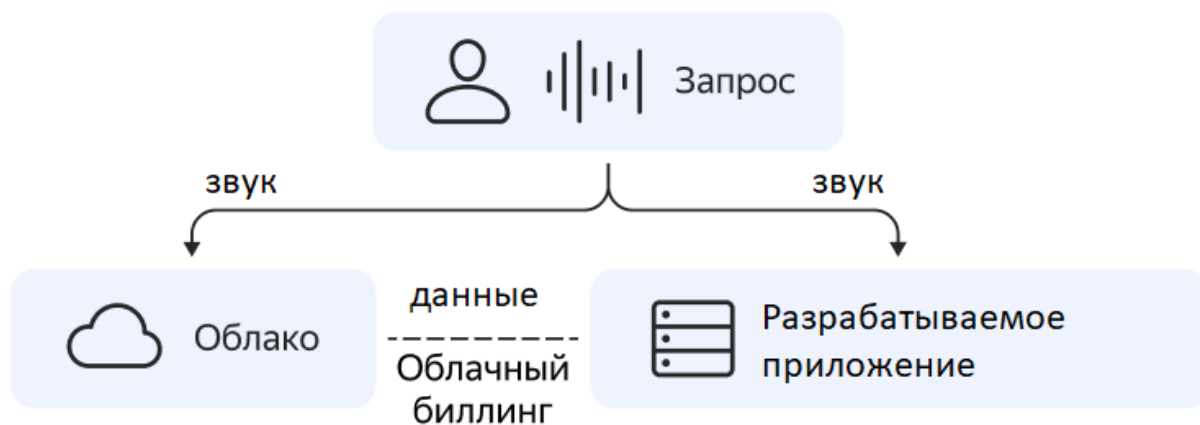


Рисунок 2 – Схема взаимодействия облачного сервиса Yandex.SpeechKit с разрабатываемым приложением

Существует возможность интеграции облачного сервиса в любое разрабатываемое приложение по средством специализированного программного интерфейса (API) [8], [11].

Разберем стандартную схему взаимодействия разрабатываемого приложения с облачным сервисом для распознавания речи с на примере Yandex.SpeechKit (рисунок 2). Когда от пользователя поступает запрос в виде речевой команды запись его голоса передается одновременно в разрабатываемое приложение и в облако с сервисом распознавания речи. Облако выполняет распознавание речи и передает результаты в виде текста в разрабатываемое приложение. Помимо текста, по запросу разрабатываемого приложения могут предаваться такие данные, как громкость голоса, длительность фразы, распознанный пол говорящего, его предполагаемый возраст и т.д. [12].

Использование облачных сервисов распознавания речи при разработке программного обеспечения обладает следующими достоинствами: высокая точность распознавания речи; простота внедрения функций распознавания речи за счет программных интерфейсов (API), предоставляемых сервисами; отсутствие необходимости разбираться в алгоритмах искусственного интеллекта, применяемых в распознавании речи.

1.2 Классификация рисков использования облачных сервисов для распознавания речи

Несмотря на удобство интеграции облачных сервисов в программные проекты, основанных на распознавание речи, их использование сопряжено с рядом рисков [15], [18]:

- риск утечки персональных данных;
- экономические риски;
- риск снижения отказоустойчивости программного обеспечения с интегрированным облачным сервисом;
- политические риски.

Рассмотрим риск утечки персональных данных. При анализе речи в облачном сервисе результат распознавания может храниться у владельцев сервиса. Анализ контекста результатов распознавания речи может быть использован для определения интересующих пользователя тем. Затем информация может предоставляться сторонним сервисам для таргетирования рекламных баннеров. В некоторых случаях облачные сервисы производят непрерывное считывание звука со встроенного в устройство микрофона. Это создает дополнительные риски случайного озвучивания персональных данных, которые не предполагалось передавать на обработку в сервис распознавания речи, например, паролей, логинов и прочее. Так же стоит отметить, что голос является одним из видов биометрических данных, которые используются, например, для идентификации клиентов банка. Поэтому передача записей голоса в сторонний сервис также связана с дополнительными рискам.

Рассмотрим экономические риски интеграции облачных сервисов распознавания речи. Практика показывает, что с ростом популярности облачных сервисов увеличивается количество его пользователей и как итог возрастает нагрузка на сервера сервиса. Если владелец облачного сервиса не планирует модернизацию аппаратной части, то для балансировки нагрузки на

сервер он производит увеличение тарифов на оказание предоставляемых услуг. Разработчики программного обеспечения, интегрирующие данный облачный сервис в свои проекты вынуждены нести дополнительные траты на оплату тарифов или перекладывать эти траты на конечных пользователей.

Рассмотрим риск снижения отказоустойчивости программного обеспечения с интегрированным облачным сервисом. Как известно, отказоустойчивость программного обеспечения зависит от работоспособности его составных частей. В случае, если в программное обеспечение интегрирован облачный сервис распознавания речи, то его работоспособность влияет на отказоустойчивость конечного приложения. При возникновении технических проблем, при принятии решения об остановке сервиса, а также при изменении программного интерфейса сервиса возможно нарушение работоспособности использующего его программного обеспечения [16], [20].

Рассмотрим политические риски, которые в настоящее время увеличивают свою актуальность. На основе своих политических убеждений зарубежные компании могут ограничить или остановить свою деятельность. При этом зарубежные компании ограничивают доступ к своим сервисам на основе географического расположения пользователей. Например, компания Intel в 2022 году ограничило для граждан России доступ к службе поддержки. В этом случае программное обеспечение, в которое интегрированы услуги облачных сервисов таких компаний, также перестанет функционировать.

Также стоит отметить, что у большинства IT компаний бизнес не ограничен одним сервисом, например помимо сервиса распознавания речи компания Google владеет видеохостингом, системой информационного поиска, социальной сетью и пр. Нарушение российских законов по одному из направлений может в итоге привести к блокированию всех или большей части сервисов, в том числе и облачного сервиса распознавания речи.

Как становится ясно из анализа рисков, перспективным направлением исследований является разработка автономного сервиса распознавания речи.

Это связано с тем, что при использовании автономного сервиса, описанные выше риски не будут возникать.

Выводы по главе 1

Первая глава посвящена анализу технологий распознавания речи.

Результаты проделанной работы позволили сделать следующие выводы:

– анализ литературных источников показал, что использование облачных сервисов распознавания речи при разработке программного обеспечения обладает следующими достоинствами: высокая точность распознавания речи, которая обусловлена постоянно пополняющейся обучающей выборкой речевых данных; простота внедрения функций распознавания речи за счет программных интерфейсов (API) предоставляемых сервисами; отсутствие необходимости разбираться в алгоритмах искусственного интеллекта применяемых в распознавании речи;

– анализ рисков использования облачных сервисов показал, что при необходимости обеспечения отказоустойчивости программного обеспечения, необходимо применение автономных систем распознавания речи.

Глава 2 Проектирование системы автономного распознавания речи

2.1 Функции системы распознавания речи

Разрабатываемая система автономного распознавания речи имеет клиент-серверную архитектуру. Распознавание речи осуществляется посредством сервера, на котором хранится языковая модель, а взаимодействие с пользователем осуществляется с помощью клиентской части системы, реализованной в виде мобильного приложения.

Разрабатываемая система включает в себя следующий функционал:

- запуск клиентского приложения вместе с операционной системой;
- распознавания пользовательских голосовых команд;
- взаимодействие с пользователем посредством вывода сообщений;
- обмен данными между с сервером, на котором хранится языковая модель;
- администрирование системы автономного распознавания речи посредством приложения.

Основной функциональной частью системы является распознавание голосовых команд. Настройка работы системы осуществляется посредством графического интерфейса приложения. Благодаря наличию функции администрирования осуществляется первичная настройка системы и корректирование параметров ее работы.

Мобильное приложение реализует следующий функционал: голосовой интерфейс, хранение настроек клиентской части системы, графический интерфейс пользователя.

Структурная схема разрабатываемой системы автономного распознавания речи представлена на рисунке 3.

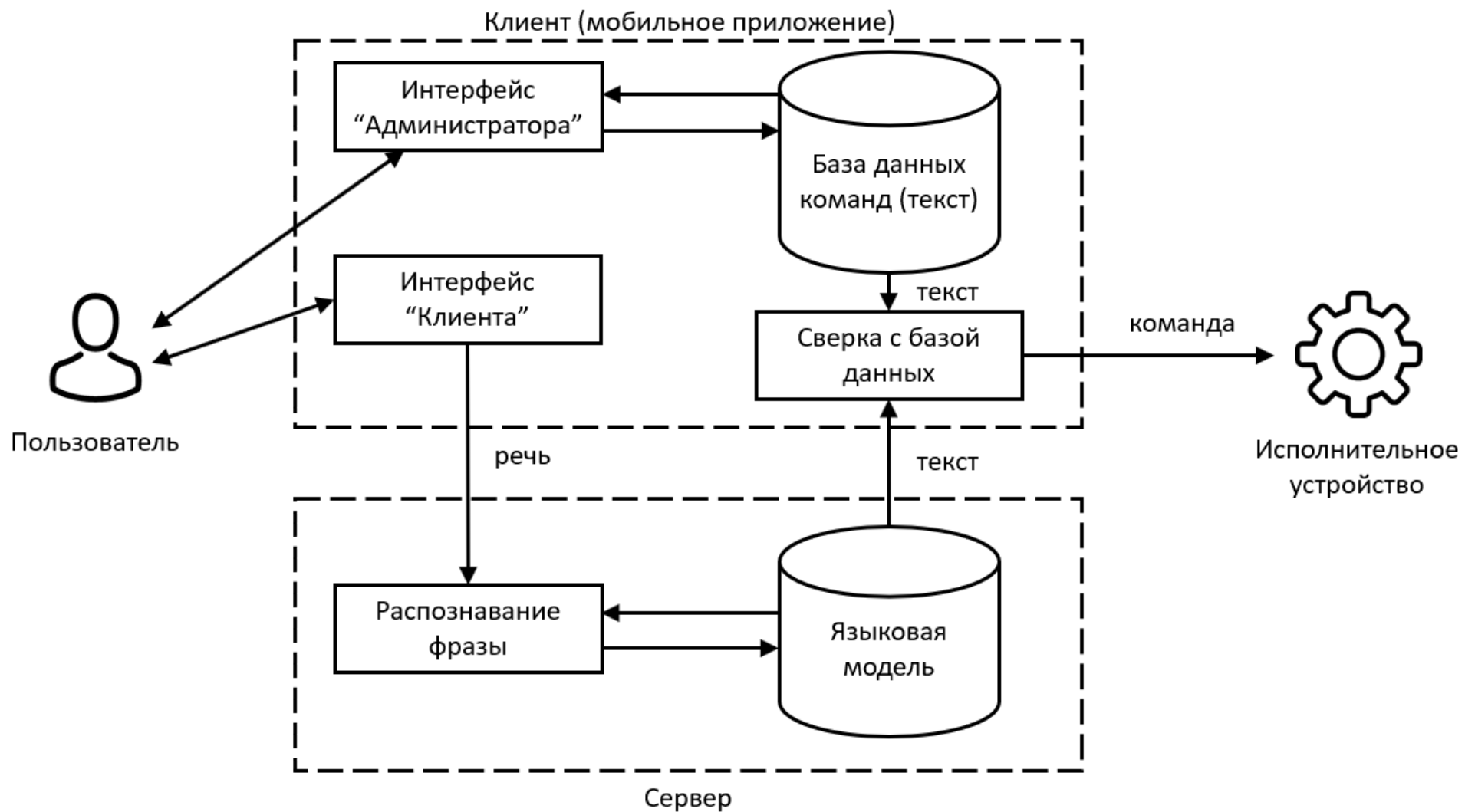


Рисунок 3 – Структурная схема системы автономного распознавания речи

2.2 Серверная часть системы распознавания речи

Для реализации серверной части системы (рисунок 3) необходимо выбрать компоненты, отвечающие за преобразование речи в текст. Можно либо самостоятельно разработать необходимые компоненты, либо воспользоваться одним из вариантов готовых решений [21], [23].

Самостоятельная разработка компонента связана с высокой трудоемкостью, поэтому предпочтительным вариантом является использование одного из готовых вариантов технологий распознавания речи. Проведем сравнительный анализ для того, чтобы определить компонент, является наиболее подходящим для серверной части разрабатываемой системы.

При выполнении сравнительного анализа использовались следующие критерии:

- расположение компонента (облако, сетевой узел или локальное расположение на устройстве);
- необходимость наличия доступа к сети Интернет для работы компонента;
- поддержка русского языка;
- поддерживаемые мобильные платформы;
- бесплатность;
- открытый исходный программный код.

При сравнительном анализе будут рассматриваться следующие компоненты:

- Yandex.SpeechKit (<https://cloud.yandex.ru/services/speechkit>);
- Google Cloud Speech API (<https://cloud.google.com/speech-to-text>);
- Alexa Voice Service (<https://developer.amazon.com>);
- SnowBoy (<https://pypi.org/project/snowboy/>).

Таблица 1 – Сравнение серверных решений для распознавания речи

Название	Расположение	Обязательный доступ к Интернет	Поддержка русского языка	Поддерживаемые мобильные платформы	Бесплатность	Открытый исходный программный код
Yandex.SpeechKit	Только в облаке Yandex	Да	Да	iOS, Android, Windows Phone	Только в образовательных целях	Нет
Google Cloud Speech API	Только в облаке Google	Да	Да	Android, iOS	Только первые 60 минут в месяц	Нет
Alexa Voice Service	Только облако Amazon	Да	Нет	Android, iOS	Нет	Нет
SnowBoy	Локально на устройстве или на выбранном сервере	Нет	Да	Android, iOS, Windows Phone	Да	Да

Результаты сравнительного анализа приведены в таблице 1.

По сочетанию особенностей наиболее подходящим компонентом для реализации серверной части автономной системы распознавания речи является библиотека SnowBoy. Данная библиотека реализует функцию распознавания речи, которая может работать локально на выбранном устройстве. В этом случае для использования SnowBoy не требуется поддерживать подключение к сети Интернет. Важной особенностью SnowBoy является поддержка русского языка и возможность работы на любых мобильных платформах (Android, iOS, Windows Phone). Библиотека является полностью бесплатной и обладает открытым исходным кодом.

2.3 Выбор протокола для обмена данными

Для реализации голосового управления устройствами необходимо выбрать протокол обмена данными с внешними устройствами. Для этого выделим наиболее часто используемые протоколы для решения данной задачи:

- MQTT (Message Queue Telemetry Transport);
- XMPP (eXtensible Messaging and Presence Protocol);
- AMQP (Advanced Message Queuing Protocol).

«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» [10].

«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 позволяет 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» [10].

«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 сообщения» [10].

На основе анализа данных из литературных источников установлено, что преимуществами протокола MQTT являются:

- проста в использовании;
- применяемый в протоколе стандарт является открытым;
- простота администрирования;
- при разработке протокола учитывалась создаваемая нагрузка сеть;
- отсутствуют ограничения на формат сообщений.

«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» [13].

На основе анализа данных из литературных источников установлено, что преимуществами протокола XMPP являются:

- использование принципов децентрализация;
- применяемый в протоколе стандарт является открытым;
- возможность изолирования XMPP сервера от внешних сетей.

Недостатком протокола XMPP является сложности при передаче

файлов, из-за которых требуется использование дополнительных проколов, таких как HTTP [24].

«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» [13].

«AMQP основан на трёх понятиях – сообщение (message), точка обмена (exchange), очередь (queue). 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. Очередь (queue) - здесь хранятся сообщения до тех пор, пока не будут забраны клиентом. Клиент всегда забирает сообщения из одной или нескольких очередей» [13].

На основе анализа данных из литературных источников установлено, что преимуществами протокола AMQP являются:

- устойчивость работы протокола в условиях нестабильной связи;
- применяемый в протоколе стандарт является открытым;
- высокая безопасность обмена сообщениями;
- надежность протокола (данный применяется в банковской сфере деятельности).

При этом главным недостатком AMQP является сложность в его реализации и обслуживании [25].

На основе проведенного анализа можно заключить, что по сочетанию достоинств и недостатков наиболее подходящий вариант для обмена

данными между системой автономного распознавания речи и внешними устройствами является протокол MQTT.

Выводы по главе 2

Вторая глава посвящена проектированию автономной системы распознавания речи.

Результаты проделанной работы позволили сделать следующие выводы:

- предложена архитектура системы автономного распознавания речи, которая основана на клиент-серверном взаимодействии: распознавание речи осуществляется посредством сервера, на котором храниться языковая модель, а взаимодействие с пользователем осуществляется с помощью клиенткой части системы, реализованной в виде мобильного приложения;

- проведен сравнительный анализ способов реализации серверной части системы (Yandex.SpeechKit, Google Cloud Speech API, Alexa Voice Service, SnowBoy) в ходе которого установлено, что библиотека SnowBoy соответствует всем необходимым требованиям: работа с русским языком, автономность, поддержка большинства мобильных платформ, открытый программный код;

- проведен сравнительный анализ протоколов для обмена данными (MQTT, XMPP и AMQP), в результате анализа для разрабатываемой системы автономного распознавания речи выбран простой в использовании и основанный на открытых стандартах протокол MQTT.

Глава 3 Разработка мобильного приложения для распознавания речи

3.1 Проектные решения, применяемые в мобильном приложении

Представленная на рисунке 3 структура автономной системы распознавания речи оптимизирована для работы с несколькими пользователями (один сервер обслуживает несколькими клиентских приложений). В рамках бакалаврской работы автономный сервис реализован в несколько упрощенной форме – и серверная и клиентская часть работают на одном мобильном устройстве на платформе Android. Поэтому разработанное мобильное программное обеспечение владеющие модули:

- модуль распознавания речи;
- модуль хранения голосовых команд;
- модуль выполнения голосовых команд пользователя.

Мобильное программное обеспечение разрабатывалось на языке Java в Android Studio. Разработанный программный код храниться в директории «SpeechRecognition»:

- файл `AppCompactPreferenceActivity.java` содержит в себе настройки проекта;
- файл `CommandHandle.java` содержит в себе методы для обработки голосовых команд пользователя;
- файл `CommandDatabase.java` содержит в себе описание базы данных для хранения команд пользователя;
- файл `SpeechRecognitionActivity.java` содержит в себе настройки графического интерфейса мобильного приложения;
- файл `SpeechRecognitionService.java` содержит в себе методы для взаимодействия с ОС Android, а также методы для работы с протоколом MQTT;

- файл `RecognitionController.java` содержит методы для взаимодействия с библиотекой `SnowBoy`;
- файл `SattingsUI.java` содержит описание интерфейса настроек приложения;
- файл `TopicsController.java` в себе методы по управлению топиками.

Для реализации технологии распознавания речи применяется библиотека `SnowBoy`. Точность распознавания речи зависит от применяемой языковой модели. В рамках выполнения бакалаврской работы использовалась готовая языковая модель для русского языка, расположенная по адресу <https://sourceforge.net/projects/cmuspinx/files/>.

Логика работы приложения показана на рисунке 4. Пользователь касается центральной части экрана при запущенном мобильном приложении и произносит в микрофон команду. Все сказанное пользователем преобразуется в текст с использованием функционала библиотеки `SnowBoy`. Затем в наборе текста ищутся ключевые слова, связанные с командами. Сверка ключевых слов производится по базе данных. В зависимости от найденного в речи ключевого слово отправляется соответствующее сообщение по протоколу MQTT. Такая логика работы позволит управлять с помощью голоса различными устройствам, например элементами система «Умного дома».

У одной и той же команды может быть несколько речевых интерпретаций («снизь яркость», «убавь свет», «сделай темней»). Поэтому все интерпретации одной и той же команды объединяются в группы, называемые топиками.

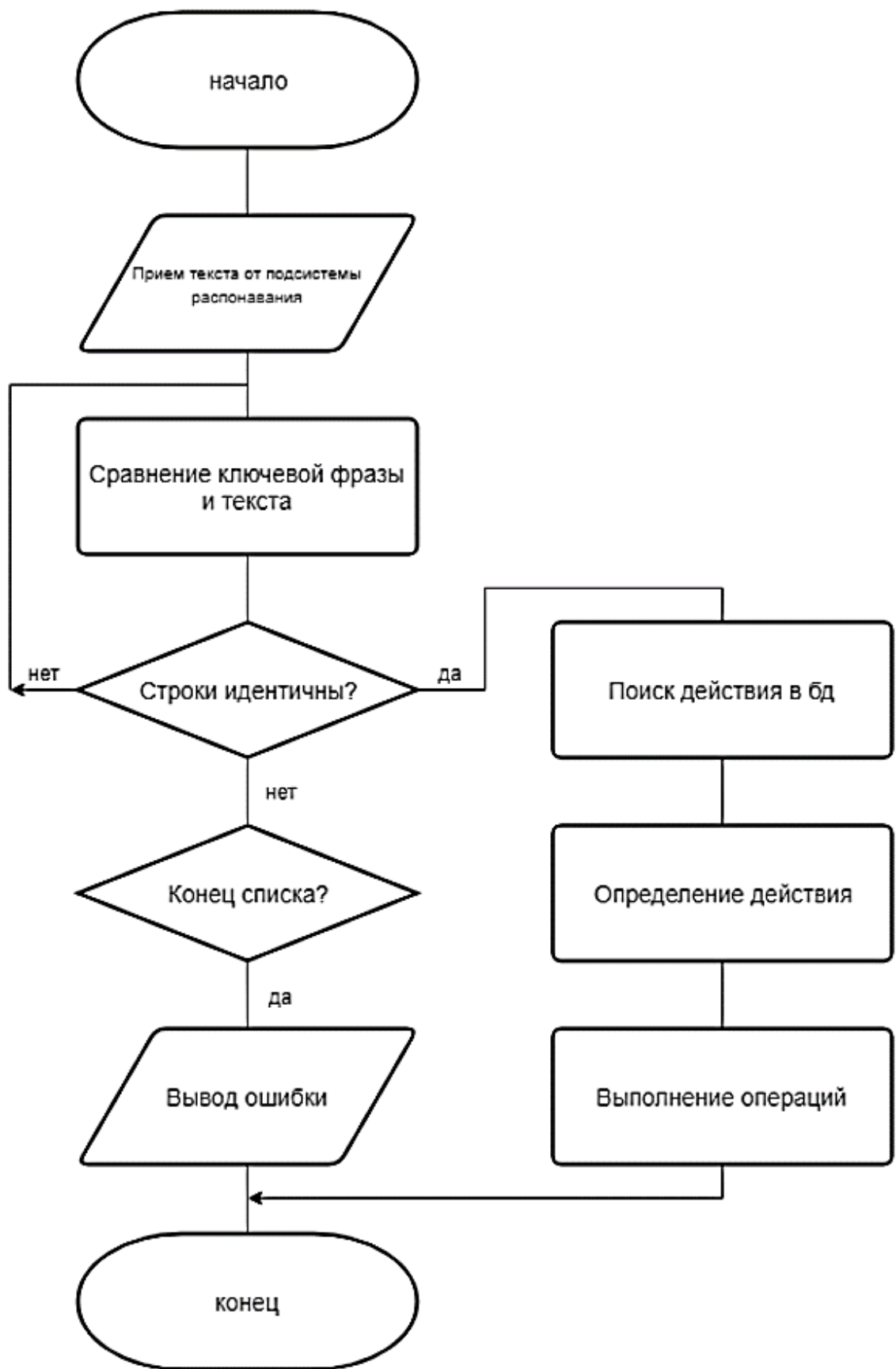


Рисунок 4 – Блок-схема сравнения распознанных фраз с базой данных команд

Для хранения топиков применяется стандартная для платформы Android база данных SQLite. Небольшая база данных хранит в себе 4 таблицы:

- mqttmessages – сообщения для обмена данными с внешними устройствами по протоколу MQTT;
- keywords – ключевые слова;
- commands – выполняемые команды;
- ttsoptions – параметры для распознавания речи.

Теперь рассмотрим особенности реализации графического интерфейса разработанного мобильного приложения.

3.2 Описание пользовательского интерфейса

Разработанное мобильное программное обеспечение обладает графическим пользовательским интерфейсом, который логически поделен на две части:

- интерфейс администратора системы;
- интерфейс клиента системы.

Интерфейс администратора системы предназначен для настройки ее работы, а интерфейс клиента системы предназначен для ее эксплуатации в обычном режиме.

В интерфейс администратора входят следующие функции:

- настройка приложения;
- настройка подключения по протоколу MQTT;
- настройка топиков.

На рисунке 5 показан сценарий использования приложения администратором системы.

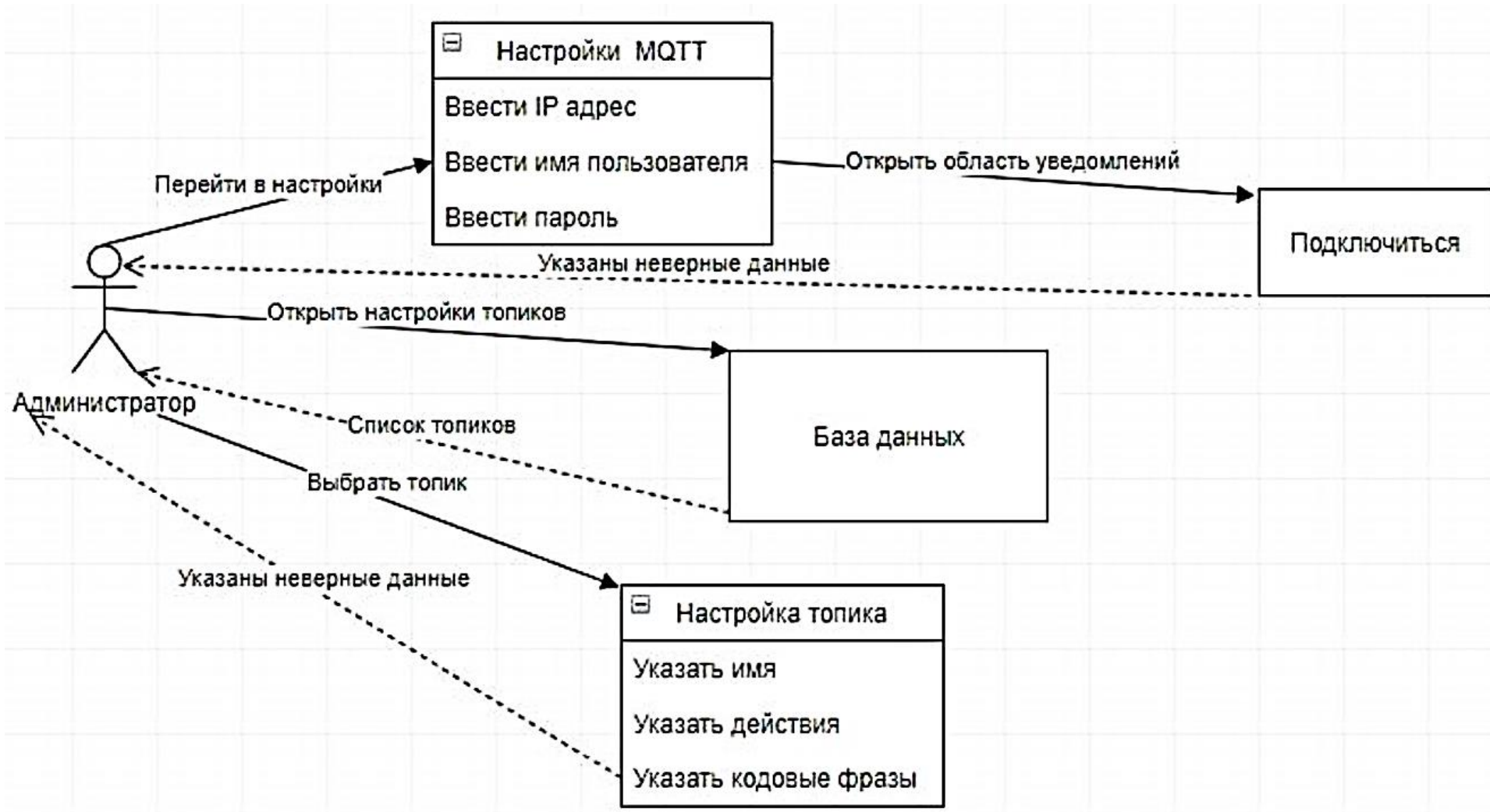


Рисунок 5 – Сценарий использования приложения администратором системы

Интерфейс администратора включает в себя:

- экран «Settings» для перемещения по основным настройкам приложения (рисунок 6);
- экран «General» для управления топиками и настройками базы данных (рисунок 7);
- экран «MQTT» для настроек передачи данных по протоколу MQTT (рисунок 8);
- экран «Recognize voice» для настроек параметров распознавания речи.

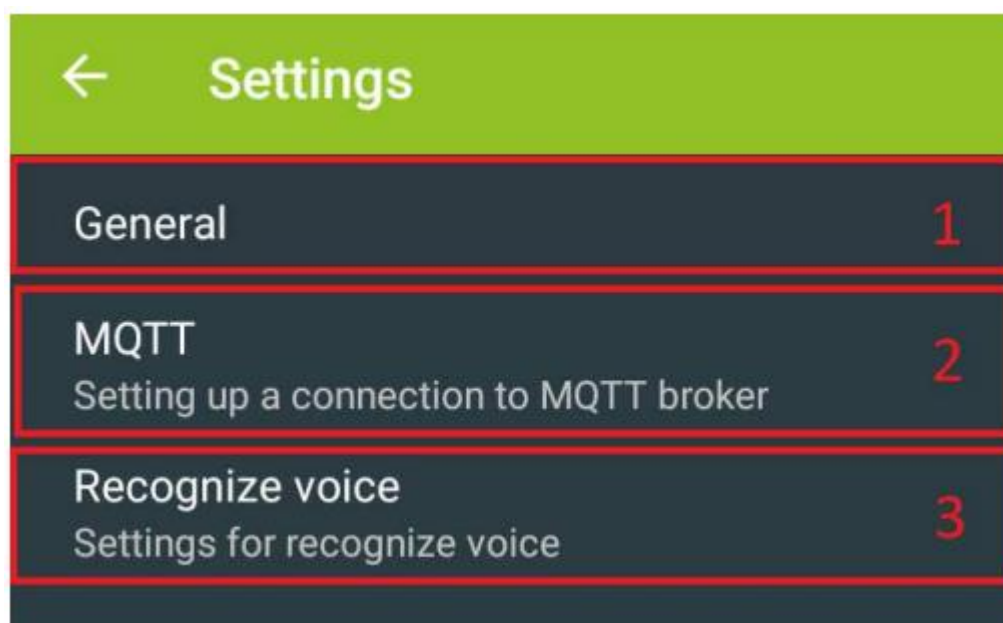


Рисунок 6 – Снимок экрана мобильного приложения с настройками Settings

На экране «Settings» доступны следующие пункты меню (рисунок 6): «General» (позиция 1), «MQTT» (позиция 2) и «Recognize voice» (3). С использованием этих пунктов можно осуществить переход к соответствующему экрану приложения и осуществить настройку работы системы автономного распознавания речи.

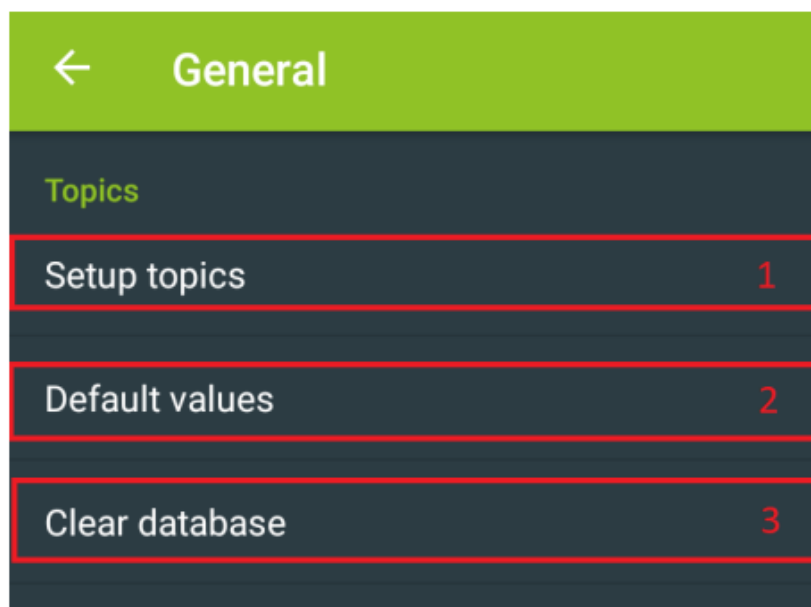


Рисунок 7 – Снимок экрана мобильного приложения с настройками General

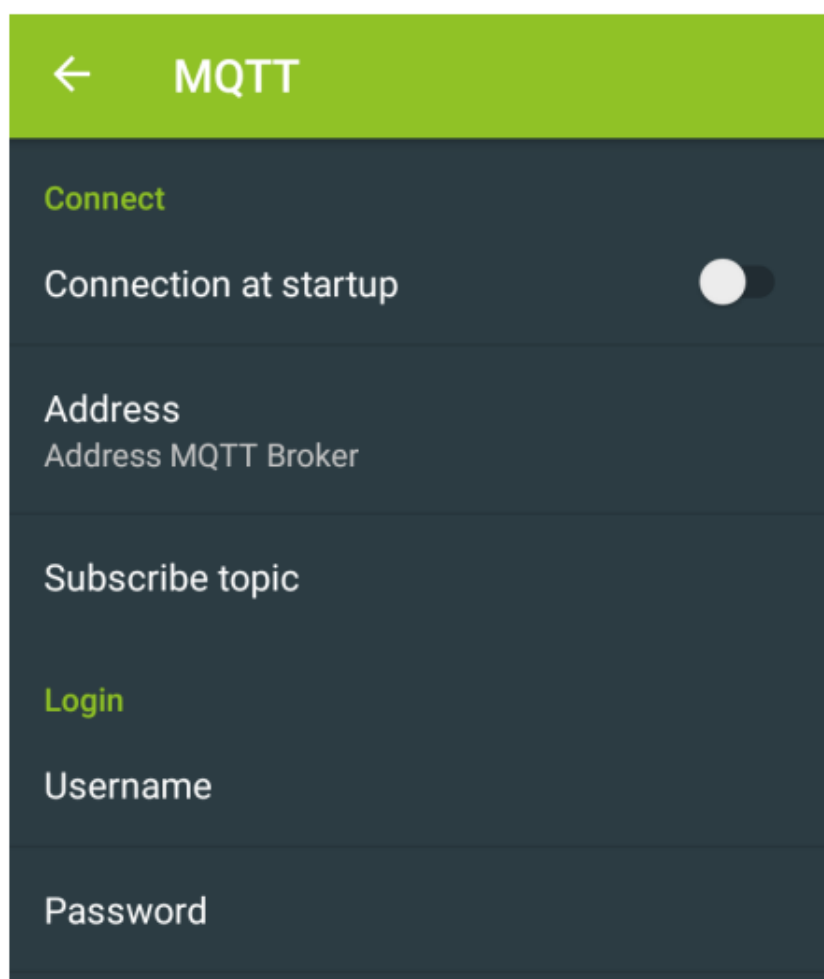


Рисунок 8 – Снимок экрана мобильного приложения с настройками работы MQTT

На экране «General» доступны следующие пункты:

- настройка топиков (позиция 1);
- сброс всех значений (позиция 2);
- очистка база данных (позиция 3).

Содержимое экрана «General» показано на рисунке 7.

На экране «MQTT» доступны следующие настройки:

- Connection at startup – запуск обмена сообщения по протоколу MQTT при старте операционной системы Android;
- Address – задание адреса MQTT брокера;
- Subscribe topic – обновление топиков;
- Username – задание имя пользователя;
- Passwor – задание пароля.

Экран «Topics» предназначенный для просмотра и редактирования топиков, хранящихся в базе данных показан на рисунке 9.

Выбор одной из записей на данном экране приведет к переходу на экран редактирования параметров топика. Пример такого экрана показан на рисунке 10, на нем пользователь работать со следующими элементами управления:

- Active (позиция 1) – выбор является ли данный топик активным;
- Dashboard (позиция 2) – выбор отображать ли данный топик на главном экране приложения;
- Topic name (позиция 3) – определение имени топика;
- Send command, TTS (позиция 4) – задание отправляемой команды и набора ключевых фраз для вызова команды.

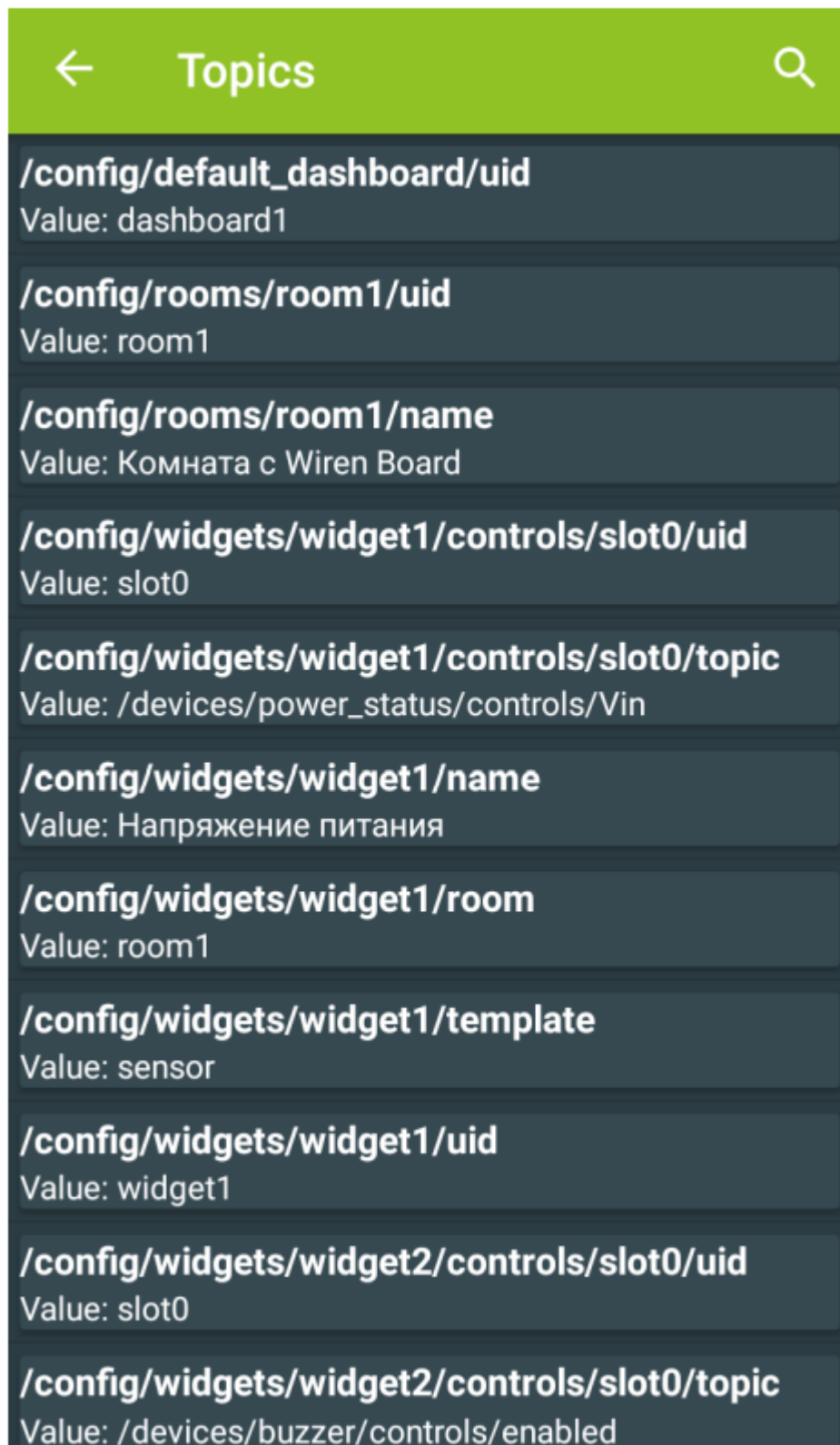


Рисунок 9 – Снимок экрана мобильного приложения с настройками Topics

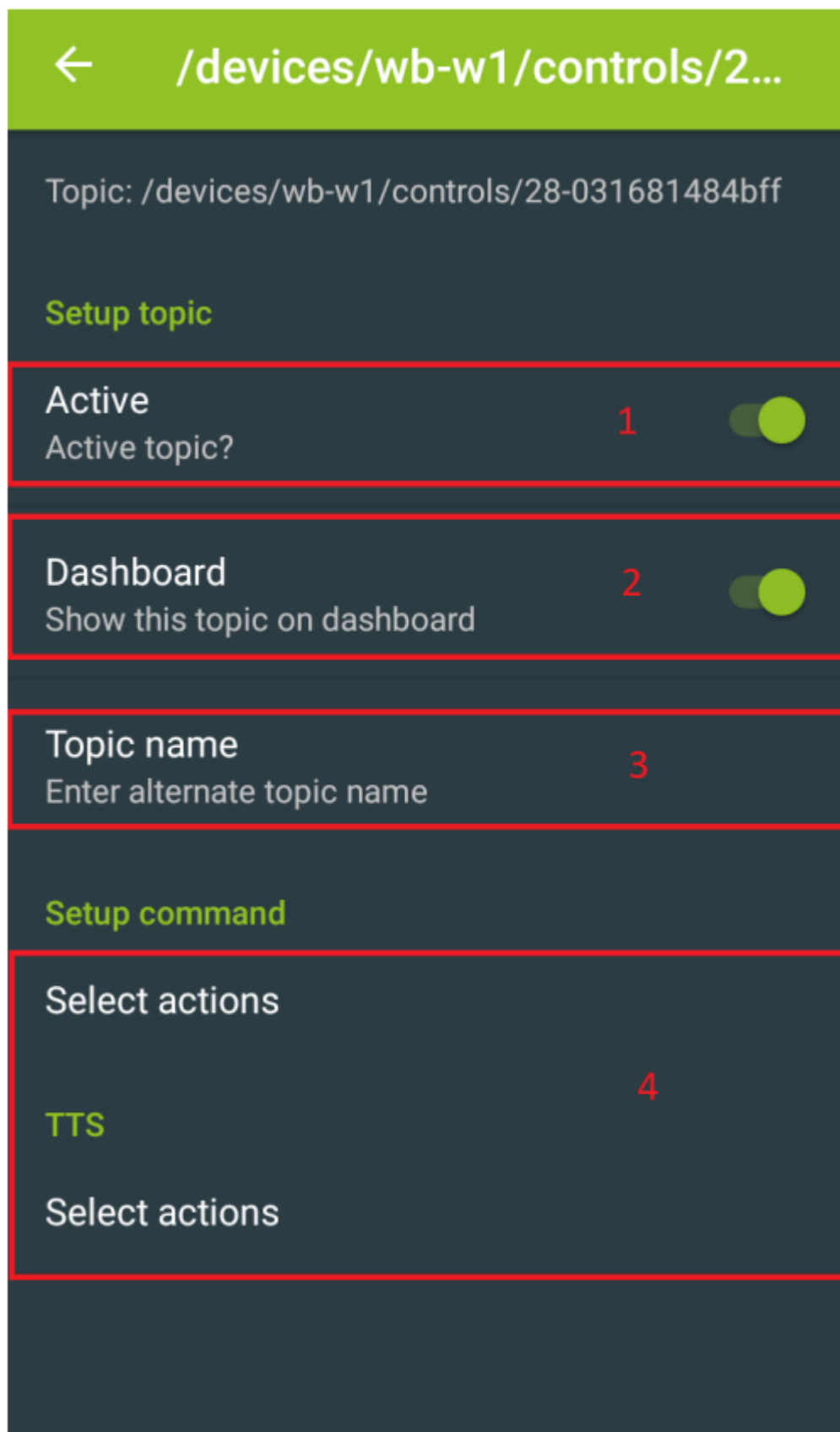


Рисунок 10 – Снимок экрана мобильного приложения с настройками экземпляра объекта Topic

При выборе на экране «General» пункта «Default values» (рисунок 7, позиция 2) откроется экран, показанный на рисунке 11. В последующем, при выборе пункта «Default ON» произойдет сброс всех настроек приложения на стандартные значения. Выбор пункта «Default OFF» приведет к отключению использования стандартных параметров системы.

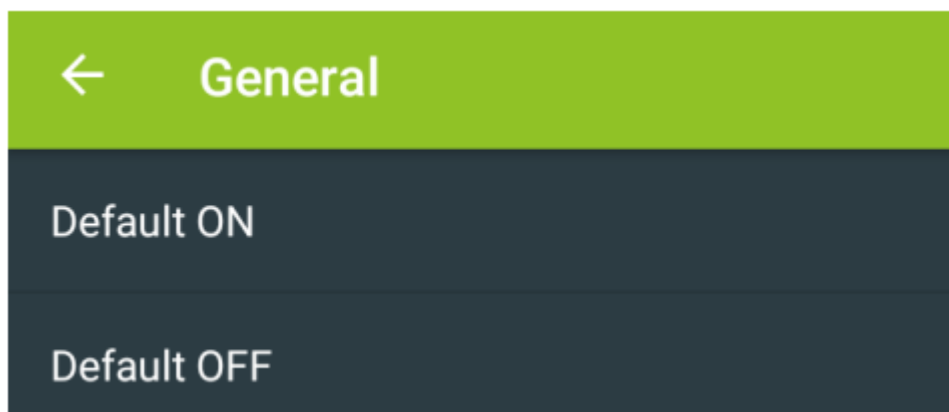


Рисунок 11 – Снимок экрана мобильного приложения с настройками General

Прейдем к рассмотрению интерфейса клиента системы. На рисунке 12 показан сценарий использования приложения клиентом системы.

В интерфейс клиента системы входят следующие элементы управления:

- настройка отображения избранных топиков;
- старт и остановка обмена сообщениями по протоколу MQTT с использованием параметров, заданных администратором системы;
- функции ввода речевых команд;
- отображение распознанных команд в интерфейсе приложения.

На рисунке 13 показан основной экран клиенту автономной системы распознавания речи. Он включает в себя 2 основных элемента управления:

- кнопка для отображения меню клиента (позиция 1);
- область для отображения избранных топиков (позиция 2).

В меню доступно 2 пункта: настройка параметров приложения и просмотр информации о мобильном приложении.

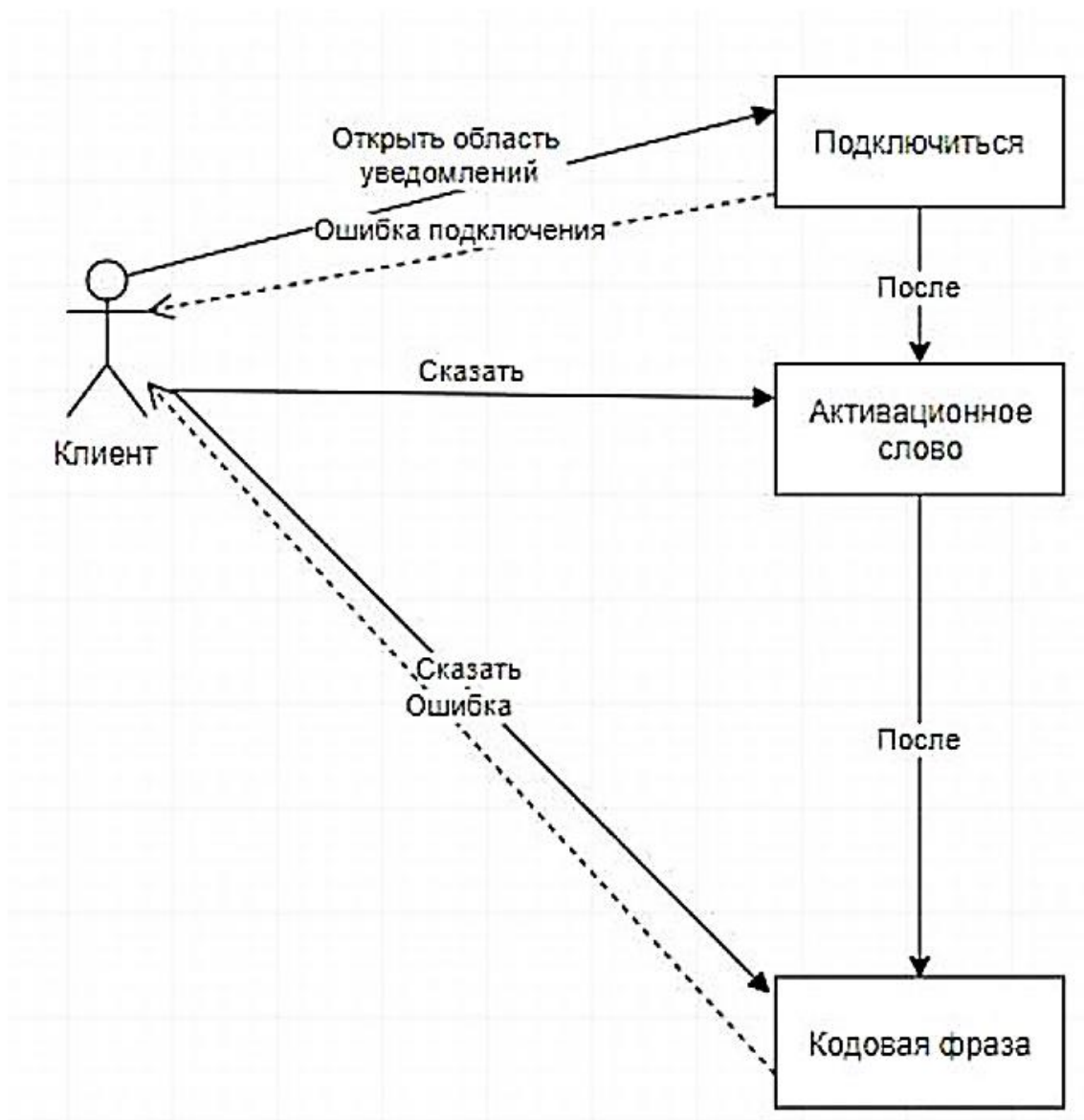


Рисунок 12 – Сценарий использования приложения клиентом системы

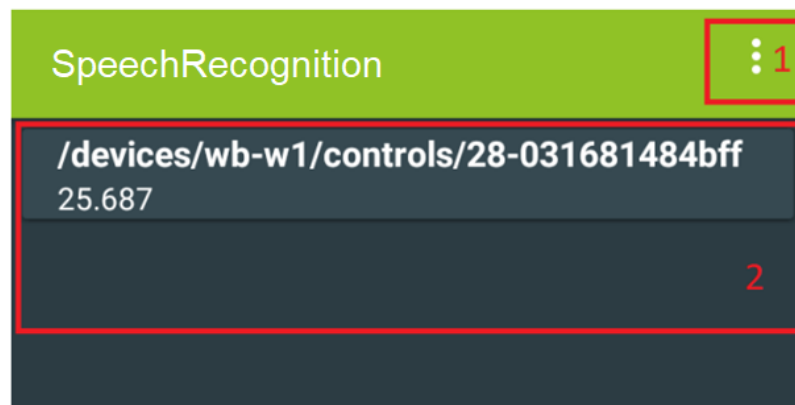


Рисунок 13 – Снимок главного экрана мобильного приложения

Интерфейс подключения представляет собой стандартные сообщения в области уведомлений ОС Android. Уведомление об отключении передачи данных по протоколу MQTT показано на рисунке 14.

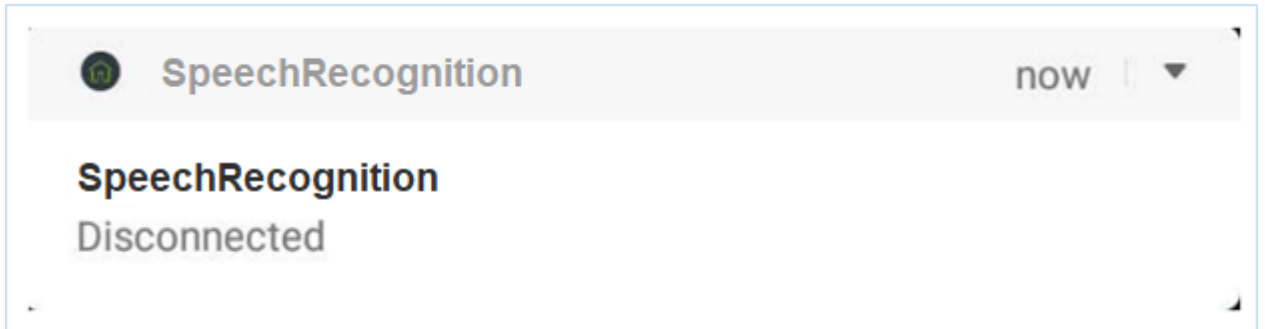


Рисунок 14 – Уведомление об отключении

Уведомление при включении передачи данных по протоколу MQTT показано на рисунке 15.



Рисунок 15 – Уведомление о подключении

Программный код основного класса разработанного мобильного приложения показан на рисунке 16.

```

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();
        }
    }
}

```

Рисунок 16 – Фрагмент программного кода основного класса

CommandHandler

Функциональное тестирование разработанного мобильного приложения показало работоспособность разработанных в бакалаврской работе проектных решений.

Выводы по главе 3

Третья глава посвящена описанию разработанного мобильного программного обеспечения.

На языке Java в среде Android studio было разработано программное обеспечение, реализующее предложенные подходы по автономному распознаванию речи. Тестирование программного обеспечения доказало состоятельность предложенных решений, возможность реализации распознавания речи без использования подключения к сети Интернет.

Заключение

В ходе выполнения бакалаврской работы были получены следующие результаты:

- анализ литературных источников показал, что использование облачных сервисов распознавания речи при разработке программного обеспечения обладает следующими достоинствами: высокая точность распознавания речи, которая обусловлена постоянно пополняющейся обучающей выборкой речевых данных; простота внедрения функций распознавания речи за счет программных интерфейсов (API) предоставляемых сервисами; отсутствие необходимости разбираться в алгоритмах искусственного интеллекта применяемых в распознавании речи;

- анализ рисков использования облачных сервисов показал, что при необходимости обеспечения отказоустойчивости программного обеспечения, необходимо применение автономных систем распознавания речи;

- предложена архитектура системы автономного распознавания речи, которая основана на клиент-серверном взаимодействии: распознавание речи осуществляется посредством сервера, на котором хранится языковая модель, а взаимодействие с пользователем осуществляется с помощью клиентской части системы, реализованной в виде мобильного приложения;

- проведен сравнительный анализ способов реализации серверной части системы (Yandex.SpeechKit, Google Cloud Speech API, Alexa Voice Service, SnowBoy) в ходе которого установлено, что библиотека SnowBoy соответствует всем необходимым требованиям: работа с русским языком, автономность, поддержка большинства мобильных платформ, открытый программный код.

- проведен сравнительный анализ протоколов для обмена данными (MQTT, XMPP и AMQP), в результате анализа для разрабатываемой системы

автономного распознавания речи выбран простой в использовании и основанный на открытых стандартах протокол MQTT;

– На языке Java в среде Android studio было разработано программное обеспечение, реализующее предложенные подходы по автономному распознаванию речи. Тестирование программного обеспечения доказало состоятельность предложенных решений, возможность реализации распознавания речи без использования подключения к сети Интернет.

Таким образом, сформулированная в бакалаврской работе цель достигнута, а также решены все поставленные задачи.

Практическая значимость работы заключается в разработке программного обеспечения, реализующего технологию автономного распознавания речи.

Список используемой литературы и используемых источников

1. Алексеев, И.В. Современные методы распознавания речи для построения голосового интерфейса управления системами специального назначения / Алексеев И.В., Митрохин М.А. // Известия вузов. Поволжский регион. Технические науки. 2019. №2 (50). URL: <https://cyberleninka.ru/article/n/sovremennye-metody-raspoznavaniya-rechi-dlya-postroeniya-golosovogo-interfeysa-upravleniya-sistemami-spetsialnogo-paznacheniya> (дата обращения: 07.06.2022).
2. Алимуратов, А.К. Алгоритм помехоустойчивой обработки речевых команд для систем голосового управления // Измерение. Мониторинг. Управление. Контроль. 2017. №1 (19). URL: <https://cyberleninka.ru/article/n/algorithm-pomehoustoychivoy-obrabotki-rechevyh-komand-dlya-sistem-golosovogo-upravleniya> (дата обращения: 07.06.2022).
3. Алимуратов, А.К. Выбор оптимального набора информативных параметров речевых сигналов для систем голосового управления / Алимуратов А.К., Чураков П.П., Тычков А.Ю. // Измерение. Мониторинг. Управление. Контроль. 2013. №1. URL: <https://cyberleninka.ru/article/n/vybor-optimalnogo-nabora-informativnyh-parametrov-rechevyh-signalov-dlya-sistem-golosovogo-upravleniya> (дата обращения: 07.06.2022).
4. Алимуратов, А.К. Исследование оптимального алгоритма обработки речевых сигналов системы голосового управления // Модели, системы, сети в экономике, технике, природе и обществе. 2015. №4 (16). URL: <https://cyberleninka.ru/article/n/issledovanie-optimalnogo-algoritma-obrabotki-rechevyh-signalov-sistemy-golosovogo-upravleniya> (дата обращения: 07.06.2022).
5. Алимуратов, А.К. Метод повышения эффективности голосового управления на основе комплементарной множественной декомпозиции на эмпирические моды / Тычков А.Ю., Зарецкий А.П., Кулешов А.П., Чураков П.П., Квитка Ю.С. // Труды МФТИ. 2017. №2 (34). URL:

<https://cyberleninka.ru/article/n/metod-povysheniya-effektivnosti-golosovogo-upravleniya-na-osnove-komplementarnoy-mnozhestvennoy-dekompozitsii-na-empiricheskie-mody> (дата обращения: 07.06.2022).

6. Алимуратов, А.К. Обзор и классификация методов обработки речевых сигналов в системах распознавания речи / Алимуратов А.К., Чураков П.П. // Измерение. Мониторинг. Управление. Контроль. 2015. №2 (12). URL: <https://cyberleninka.ru/article/n/obzor-i-klassifikatsiya-metodov-obrabotki-rechevyh-signalov-v-sistemah-raspoznavaniya-rechi> (дата обращения: 07.06.2022).

7. Алимуратов, А.К. Оптимальный алгоритм обработки речевых команд для системы голосового управления // Модели, системы, сети в экономике, технике, природе и обществе. 2015. №2 (14). URL: <https://cyberleninka.ru/article/n/optimalnyy-algoritm-obrabotki-rechevyh-komand-dlya-sistemy-golosovogo-upravleniya> (дата обращения: 07.06.2022).

8. Астапов, С.С. Автоматическое распознавание речи в условиях шума музыки на многоканальных записях с удаленного микрофона / Шуранов Е.В., Лаврентьев А.В., Кабаров В.И. // Научно-технический вестник информационных технологий, механики и оптики. 2019. №3. URL: <https://cyberleninka.ru/article/n/avtomaticheskoe-raspoznavanie-rechi-v-usloviyah-shuma-muzyki-na-mnogokanalnyh-zapisyah-s-udalennogo-mikrofona> (дата обращения: 07.06.2022).

9. Бердибаева, Г.К., Применение искусственных нейронных сетей для распознавания речевых команд / Бердибаева Г.К., Бодин О.Н., Громков Н.В., Козлов В.В., Ожикенов К.А., Пижонков Я.А. // Измерение. Мониторинг. Управление. Контроль. 2017. №2 (20). URL: <https://cyberleninka.ru/article/n/primenenie-iskusstvennyh-neyronnyh-setey-dlya-raspoznavaniya-rechevyh-komand> (дата обращения: 07.06.2022).

10. Дикий, Д.И. Протокол передачи данных MQTT в модели удаленного управления правами доступа для сетей интернета // Научно-технический вестник информационных технологий, механики и оптики.

2019. №1. URL: <https://cyberleninka.ru/article/n/protokol-peredachi-dannyh-mqtt-v-modeli-udalennogo-upravleniya-pravami-dostupa-dlya-setey-interneta> (дата обращения: 07.06.2022).

11. Меденников, И.П. Двухэтапный алгоритм инициализации обучения акустических моделей на основе глубоких нейронных сетей // Научно-технический вестник информационных технологий, механики и оптики. 2016. №2. URL: <https://cyberleninka.ru/article/n/dvuhetapnyy-algoritm-initsializatsii-obucheniya-akusticheskikh-modeley-na-osnove-glubokih-neyronnyh-setey> (дата обращения: 07.06.2022).

12. Муратов, Н.А. Основные методы обработки речевых сообщений // Новые информационные технологии в автоматизированных системах. 2018. №21. URL: <https://cyberleninka.ru/article/n/osnovnyye-metody-obrabotki-rechevyh-soobscheniy> (дата обращения: 07.06.2022).

13. Селезнёв, С.П. Архитектура промышленных приложений IoT и протоколы amqp, mqtt, jms, rest, CoAP, xmpp, dds // International Journal of Open Information Technologies. 2019. №5. URL: <https://cyberleninka.ru/article/n/arhitektura-promyshlennyh-prilozheniy-iot-i-protokoly-amqp-mqtt-jms-rest-coap-xmpp-dds> (дата обращения: 07.06.2022).

14. Тампель, И.Б. Автоматическое распознавание речи - основные этапы за 50 лет // Научно-технический вестник информационных технологий, механики и оптики. 2015. №6. URL: <https://cyberleninka.ru/article/n/avtomaticheskoe-raspoznavanie-rechi-osnovnye-etapy-za-50-let> (дата обращения: 07.06.2022).

15. Шматков, В.Н. Взаимодействие с устройствами интернета вещей с использованием голосового интерфейса / Бонковски П., Медведев Д.С., Корзухин С.В., Голендухин Д.В., Спыну С.Ф., Муромцев Д.И. // Научно-технический вестник информационных технологий, механики и оптики. 2019. №4. URL: <https://cyberleninka.ru/article/n/vzaimodeystvie-s-ustroystvami-interneta-veschey-s-ispolzovaniem-golosovogo-interfeysa> (дата обращения: 07.06.2022).

16. Haripriya, A. P. Secure-MQTT: an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things / Haripriya A. P., Kulothungan K. // URASIP Journal on Wireless Communications and Networking, 2019. - №90. – pp. 10-25.
17. Vytautas, R. On the Adaptation of Foreign Language Speech Recognition Engines for Lithuanian Speech Recognition / Vytautas R., Rytis M., Algimantas R. Kastytis R. // BIS 2009: Business Information Systems Workshops, 2009. – №37. – pp. 113-118.
18. Xin, M. Speech Feature Extraction Based on Wavelet Modulation Scale for Robust Speech Recognition / Xin Ma, Weidong Zhou, Fang Ju, Qi Jiang // ICONIP 2006: Neural Information Processing, 2006. – №42. – pp. 499–505.
19. Piotr, K. Speech Recognition Based on Open Source Speech Processing Software / Piotr K., Adam D., Jacek I., Jan K., Jacek Ś. // International Conference on Computer Networks CN 2014: Computer Networks, 2014. – №43. – pp. 308–317.
20. Ramana, A.V. Investigation of Speech Coding Effects on Different Speech Sounds in Automatic Speech Recognition / A.V. Ramana, P. Laxminarayana, P. Mythilisharan // Indo-Japanese Conference on Perception and Machine Intelligence PerMIn 2012: Perception and Machine Intelligence, 2012. – № 71. – pp. 367–377.