

В докладе рассказывается о реализации функции голосового ввода в многопользовательском чате с использованием языка программирования Java и сервиса распознавания голоса Google API. В методе onCreate определяется кнопка для запуска голосового ввода информации. Алгоритм работает следующим образом. Берется id данной кнопки, после чего создается метод setOnClickListener, в котором создается новое намерение Intent для вызова метода ACTION_RECOGNIZE_SPEECH. Этот метод нужен для запуска действия, которое запрашивает у пользователя речь и отправляет ее через распознаватель речи. После чего с помощью EXTRA_LANGUAGE_MODEL и LANGUAGE_MODEL_FREE_FORM идет распознавание языка, на котором говорит пользователь. При этом отображается подсказка, с помощью которой можно понять, что уже можно говорить. Это делается с помощью EXTRA_PROMPT. Далее вызывается переопределенный метод onActivityResult, в котором благодаря EXTRA_RESULTS формируется ArrayList<String> результатов распознавания, то есть текста, который сказал пользователь. Текст появляется в поле editText, после чего его можно отправить участникам чата.

Литература

1 Грожек, Г. В. Разработка многопользовательского чата с использованием инструментов FIREBASE / Г. В. Грожек // XXVII Республиканская научная конференция студентов и аспирантов «Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях», 2024 г., Гомель: ГГУ им. Ф. Скорины. – С. 178.

И. Н. Громыко, Е. А. Ружицкая
(ГГУ имени Ф. Скорины, Гомель)

РАЗРАБОТКА БАЗЫ ДАННЫХ WEB-ПРИЛОЖЕНИЯ ДЛЯ ПЛАНИРОВАНИЯ ЗАДАЧ „MY PLANNER: TASK MANAGER“

При разработке любого программного средства, нацеленного на автоматизацию управления хранимыми данными, неотъемлемой ча-

стью является создание базы данных. Она должна быть не только исчерпывающей, структурированной, минимизированной, но и эффективной с точки зрения хранения и обработки информации, а также нормализованной. Нормализация позволяет избавиться от избыточности данных и несогласованной зависимости. Хорошей практикой в программировании считается предварительное моделирование структуры и логики разрабатываемой системы.

Для создания модели базы данных в данной работе используется методология IDEF1X, которая позволяет четко определить сущности, их атрибуты и связи между ними, что обеспечивает создание логической и физической структуры базы данных. Методология IDEF1X выделяет три уровня представления модели данных.

Первый уровень – уровень сущностей и связей. На этом уровне можно выделить следующие сущности: «Пользователь», «Проект», «Задача» и «Метка». Для каждой сущности можно выделить свои атрибуты. Сущности тесно связаны между собой, однако точную связь тяжело установить по причине того, что сущности «Метка» и «Задача» имеют связь типа «многие ко многим». После добавления первичных ключей модель переходит ко второй нормальной форме и второму уровню представления – уровню ключей. Для доведения схемы до третьего уровня – уровня всех атрибутов – необходимо ввести внешние ключи. Однако это невозможно сделать, пока между сущностями имеются отношения «многие ко многим», так как это противоречит первой нормальной форме в реляционной модели данных. Чтобы привести такие связи к типу «один ко многим», нужно добавить промежуточные сущности «МеткаЗадачи». Теперь сущности могут быть связаны между собой с помощью внешних ключей. Введены дополнительные таблицы для хранения подробных данных о пользователях, задачах, проектах и прогрессе выполнения. Это приводит модель базы данных к третьей нормальной форме. После внесения всех атрибутов структура базы данных считается полностью определенной и готовой к реализации.