

Для управления ассортиментом без необходимости ручного доступа к базе данных была реализована административная панель внутри самого бота. Авторизованные администраторы могут добавлять новые товары, редактировать информацию о существующих позициях, а также удалять устаревшие или недоступные товары. В целях безопасности данная функция доступна только пользователям с определённым `user_id`, внесённым в список администраторов. Ввод данных при добавлении и редактировании реализован пошагово с использованием FSM, что позволяет избежать ошибок при вводе и делает процесс максимально удобным.

Также была реализована система категорий, которая структурирует ассортимент магазина и облегчает пользователям поиск нужных товаров. При взаимодействии с ботом пользователь сначала выбирает интересующую категорию, после чего получает список товаров, относящихся к данной группе. Категории отображаются в виде кнопок, что делает интерфейс более визуально удобным и современным. Внутри базы данных для каждого товара предусмотрено поле `category`, что позволяет гибко управлять структурой ассортимента и легко добавлять новые группы товаров.

Созданный многофункциональный Телеграм-бот имеет интуитивно понятный пользовательский интерфейс. Бот включает в себя структуру хранения данных, что значительно упрощает доступ к информации о товарах и позволяет быстро обновлять ассортимент. Кроме того, бот обеспечивает удобство управления заказами и может служить основой для более сложных проектов.

Одним из возможных направлений развития такого бота является интеграция с платежными системами, что позволяет пользователям оформлять и оплачивать заказы прямо в мессенджере. Это открывает новые горизонты для автоматизации процессов интернет-магазина и улучшения пользовательского опыта, позволяя делать покупки более удобными и быстрыми.

Практическая значимость разработки заключается в том, что бот предоставляет удобный инструмент для управления покупками, а также демонстрирует основы работы с библиотекой `Aiogram`. Это делает его полезным не только для конечных пользователей, которые могут легко получать информацию о товарах и оформлять заказы, но и для разработчиков, которые изучают создание Телеграм-ботов.

Полученные знания и опыт могут быть использованы для разработки более сложных систем автоматизации в различных областях, включая, но не ограничиваясь, сферой услуг, образования и развлечений.

Литература

1 Aiogram 3.19.0 documentation: [сайт]. – aiogram Team, 2025. – URL: <https://docs.aiogram.dev/en/v3.19.0/#> (дата обращения: 24.03.2025).

2 SQLite documentation: [сайт]. – 2025. – URL: <https://www.sqlite.org/docs.html> (дата обращения: 24.03.2025).

УДК 004.522:004.62

В. А. Шкарубо

ИНТЕГРАЦИЯ РАСПОЗНАВАНИЯ РЕЧИ И ОБЛАЧНЫХ СЕРВИСОВ В НАСТОЛЬНЫХ ПРИЛОЖЕНИЯХ ДЛЯ ОБРАБОТКИ МЕДИЦИНСКИХ ДАННЫХ

В статье рассматривается опыт разработки настольного приложения для работы с медицинскими данными, в котором реализованы технологии распознавания речи и облачного хранения. Приводится архитектура программного решения, описываются

принципы интеграции речевого интерфейса с облачной базой данных Firebase, а также механизмы редактирования, удаления и синхронизации записей. Отдельное внимание уделено проблемам устойчивости соединения, обработке ошибок и вопросам гибкости пользовательского взаимодействия. Предложенное решение позволяет значительно упростить процесс ввода информации, обеспечить её надёжность и расширить доступ к медицинским данным.

Современные медицинские системы всё чаще нуждаются в решениях, способных обеспечивать быструю и надёжную обработку данных пациентов. Особенно актуальной становится задача интеграции технологий голосового ввода с облачными сервисами хранения и синхронизации информации [1].

Распознавание речи позволяет существенно упростить процедуру ввода данных, а использование облачных решений обеспечивает централизованное хранение и доступ к информации с разных устройств. Совмещение этих двух подходов даёт возможность создавать современные настольные приложения, соответствующие требованиям медицинской практики.

Целью данной работы стало создание настольного приложения, обеспечивающего эффективное управление медицинскими данными с использованием распознавания речи и облачного хранилища Firebase. В процессе разработки были реализованы функции добавления, редактирования, удаления и синхронизации записей, обеспечена возможность переключения между локальным и облачным режимами хранения, а также внедрена система оповещения об ошибках.

Архитектура программного решения. Созданное приложение представляет собой настольную систему, предназначенную для записи и обработки информации о пациентах. Пользовательский интерфейс разработан с использованием библиотеки Tkinter, что обеспечивает интуитивно понятное и удобное взаимодействие. Пользователь может вводить данные как вручную, так и голосом, используя встроенную систему распознавания речи, реализованную на базе библиотеки SpeechRecognition.

Архитектура приложения включает несколько основных компонентов. Интерфейс взаимодействия с пользователем служит центральной точкой доступа ко всем функциям программы (рисунок 1). Модуль распознавания речи отвечает за преобразование аудиопотока в текстовую форму, которую можно сохранить в базе данных. Модуль синхронизации с Firebase обеспечивает загрузку и выгрузку данных в облачное хранилище. Также реализовано локальное хранение с помощью CSV-файлов, позволяющее сохранять и загружать данные на устройстве пользователя без подключения к интернету. Механизмы обработки ошибок и уведомлений помогают пользователю своевременно реагировать на проблемы с сетью или доступом к данным.

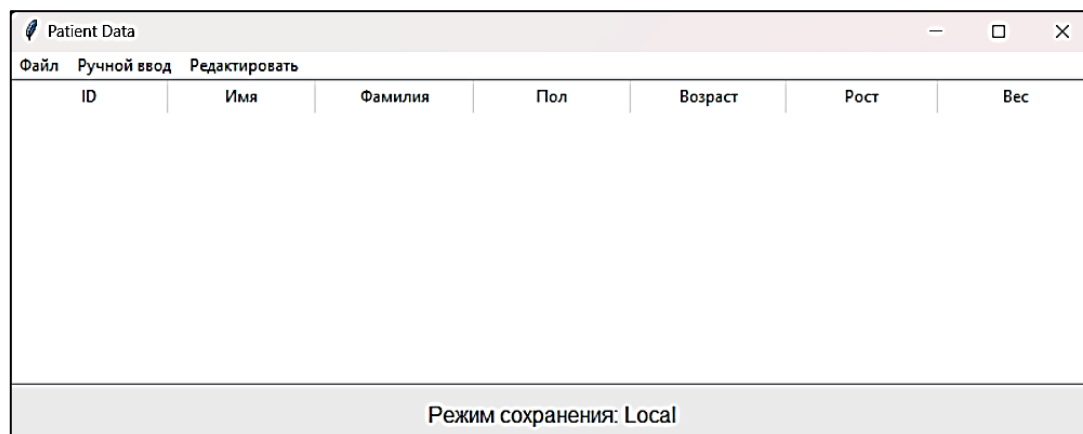


Рисунок 1 – Интерфейс приложения

Актуальность интеграции речевого ввода и облачных сервисов. Быстрый и точный ввод цифровой информации становится всё более важным в различных отраслях. Использование речевого интерфейса в сочетании с облачными технологиями позволяет значительно ускорить этот процесс, повысить его надёжность и доступность.

В медицине оперативное внесение данных о пациентах критично для своевременного принятия решений. Медицинский персонал может использовать голосовой ввод для записи симптомов, истории болезни и других сведений, что ускоряет рабочий процесс и снижает вероятность ошибок.

В образовательной сфере преподаватели могут с помощью голосовых команд регистрировать посещаемость, оценки и другую информацию, повышая эффективность административной работы и улучшая планирование учебного процесса.

В бизнесе и финансах своевременное обновление информации с помощью голосового ввода позволяет быстрее реагировать на изменения в рыночной ситуации, вносить корректировки в отчётность и принимать обоснованные решения. Облачные базы данных обеспечивают совместную работу сотрудников в реальном времени.

В индустрии развлечений голосовой ввод позволяет оперативно управлять контентом – будь то создание метаданных, диктовка описаний или организация медиатеки. Облачное хранение обеспечивает доступ к контенту из любой точки мира.

В логистике и транспорте использование голосового интерфейса помогает регистрировать перемещение грузов, статус поставок и другие параметры без необходимости ручного ввода. Это особенно актуально в условиях работы на ходу или при высокой плотности операций.

Таким образом, речевые интерфейсы и облачные технологии обеспечивают не только удобство, но и значительное повышение эффективности в работе с цифровыми данными. Их интеграция позволяет автоматизировать рутинные задачи, сократить издержки и повысить качество обслуживания.

Функциональные возможности. Пользователь может выбрать режим работы приложения – локальный или облачный. В первом случае все данные сохраняются на локальное устройство, что удобно при отсутствии подключения к интернету. Во втором – используется база данных Firebase, обеспечивающая синхронизацию и централизованный доступ к данным.

Ввод данных реализован как через формы, так и через голосовые команды (рисунок 2). Последние преобразуются в текст с помощью библиотеки SpeechRecognition, а затем автоматически заполняют поля формы. Это позволяет значительно ускорить процесс ввода информации и минимизировать ошибки, связанные с ручным вводом.

Patient Data								
Файл Ручной ввод Редактировать								
	ID	Имя	Фамилия	Пол	Возраст	Рост	Вес	
1		Ирина	Романова	женский	52	160	64	
2		Иван	Смирнов	мужской	54	189	98	
4		Арина	Гулевич	женский	20	168	53	
5		Евгений	Лапин	мужской	25	184	100	
6		Павел	Иванов	мужской	64	174	58	
7		Олег	Соболев	мужской	43	170	89	

Режим сохранения: Firebase

```

Установлена фамилия: Соболев
Данные пациента 7 сохранены в Firebase: {'First Name': 'Oler', 'Last Name': 'Соболев', 'Gender': None, 'Age': None, 'Height': None, 'Weight': None}
Данные пациента 7 сохранены в Firebase: {'First Name': 'Oler', 'Last Name': 'Соболев', 'Gender': None, 'Age': None, 'Height': None, 'Weight': None}
Установлен пол: мужской
Данные пациента 7 сохранены в Firebase: {'First Name': 'Oler', 'Last Name': 'Соболев', 'Gender': 'мужской', 'Age': None, 'Height': None, 'Weight': None}
Установлен возраст: 43
Данные пациента 7 сохранены в Firebase: {'First Name': 'Oler', 'Last Name': 'Соболев', 'Gender': 'мужской', 'Age': 43, 'Height': None, 'Weight': None}
Данные пациента 7 сохранены в Firebase: {'First Name': 'Oler', 'Last Name': 'Соболев', 'Gender': 'мужской', 'Age': 43, 'Height': None, 'Weight': None}
Установлен рост: 170
Данные пациента 7 сохранены в Firebase: {'First Name': 'Oler', 'Last Name': 'Соболев', 'Gender': 'мужской', 'Age': 43, 'Height': 170, 'Weight': None}
Установлен вес: 89
Данные пациента 7 сохранены в Firebase: {'First Name': 'Oler', 'Last Name': 'Соболев', 'Gender': 'мужской', 'Age': 43, 'Height': 170, 'Weight': 89}

```

Рисунок 2 – Пример ввода данных

Также реализована возможность редактирования и удаления записей. При двойном щелчке на записи открывается окно редактирования, где можно изменить любые поля. Удаление записи сопровождается подтверждением действия, что исключает случайные удаления. Вся информация обновляется в реальном времени, отражаясь как в интерфейсе, так и в базе данных.

Процесс ввода цифровой информации был тщательно протестирован и оптимизирован для обеспечения его точности. Получение обратной связи от пользователей помогло выявить и устранить возможные проблемы.

Сценарии использования и преимущества. Предлагаемое приложение может использоваться в медицинских учреждениях для регистрации данных пациентов, истории болезней, параметров здоровья и других сведений. Голосовой ввод особенно полезен в условиях ограниченного времени или невозможности ручного ввода. Облачное хранилище позволяет нескольким специалистам получать доступ к актуальной информации с различных устройств и в разных локациях.

Преимуществом решения является также его адаптируемость: возможность выбора режима хранения, гибкая архитектура, простота расширения функционала. Реализация интерфейса на Tkinter делает приложение кроссплатформенным в пределах настольной среды Python.

Особое внимание было уделено обеспечению устойчивой работы системы в условиях нестабильного интернет-соединения. При возникновении ошибок подключения к Firebase пользователь получает уведомление с возможными способами устранения проблемы. В случае невозможности выполнения операции приложение автоматически переключается на локальный режим, что исключает потерю данных.

Также были реализованы дополнительные проверки, позволяющие избежать дублирования записей и ошибок формата. Все операции сопровождаются обратной связью в интерфейсе, повышающей информативность и удобство работы.

Разработанное приложение продемонстрировало эффективность интеграции распознавания речи и облачных сервисов для задач медицинского документооборота. Совмещение голосового ввода и гибких механизмов хранения позволяет ускорить процесс ввода данных, повысить их точность и упростить доступ к информации.

Литература

1 Ронжин, А. Л. Речевой и многомодальный интерфейсы / А. Л. Ронжин, А. А. Карпов, И. В. Ли. – М. : Наука, 2006. – 173 с.