

DOI: <https://doi.org/10.36910/6775-2524-0560-2024-55-32>

УДК 004.75

Якобчук Богдан Анатолійович, здобувач вищої освіти
кафедри комп'ютерної інженерії та кібербезпеки

<https://orcid.org/0009-0002-6979-725X>

Христинець Наталія Анатоліївна, к.т.н., доцент

<https://orcid.org/0000-0002-4836-7632>

Луцький національний технічний університет, м. Луцьк, Україна

СТВОРЕННЯ ПРОСТОГО ГОЛОСОВОГО ПОМІЧНИКА У ДОДАТКУ ДЛЯ ПЛАТФОРМИ ANDROID

Якобчук Б.А., Христинець Н.А. Створення простого голосового помічника у додатку для платформи **Android**. У статті представлено вичерпний посібник зі створення простого голосового помічника для Android-додатків. Голосовий помічник, названий «ExpertAvon», призначений для надання користувачам функціональностей взаємодії з додатком, у який інтегрований голосовий помічник або з іншими додатками чи компонентами смартфона. Процес розробки передбачає інтеграцію технології розпізнавання мовлення та подальшу реакцію на отримані голосові команди. Стаття містить покрокову інструкцію зі створення ExpertAvon, включаючи налаштування інструментів, бібліотек, реалізацію перетворення аудіо в текст та подальшу взаємодію з компонентами додатку.

Ключові слова: голосовий помічник, Android-додаток, розпізнавання мовлення, ExpertAvon, інтеграція голосових команд, взаємодія з додатками, перетворення аудіо в текст, розробка мобільних додатків.

Yakobchuk B., Khrystynets N. **Creating a simple voice assistant in an application for the Android platform.** The article presents a comprehensive guide to creating a simple voice assistant for Android applications. The voice assistant, named «ExpertAvon», is designed to provide users with the functionality of interacting with the application into which the voice assistant is integrated or with other applications or components of the smartphone. The development process involves the integration of speech recognition technology and the subsequent response to received voice commands. The article contains step-by-step instructions for creating ExpertAvon, including setting up tools, libraries, implementing audio-to-text conversion, and further interaction with application components.

Keywords: voice assistant, Android application, speech recognition, ExpertAvon, voice command integration, interaction with applications, audio to text conversion, mobile application development.

Постановка наукової проблеми. Проблемою дослідження є вирішення ряду ключових питань у розробці простого голосового помічника в середовищі Android Studio засобами мови програмування Java. Ці дослідження спрямовані на створення процесу розпізнавання голосових команд, через перетворення аудіо в текст.

Проблему створення оптимального функціоналу перетворення аудіо даних в текстові дані, було вирішено через впровадження у додаток хмарного сервісу розпізнавання мовлення Google Speech Recognition API.

Завдання дослідження полягає у вирішенні технічних та концептуальних аспектів створення голосових помічників, а також ознайомлення із процесом реалізації функції розпізнавання мовлення, шляхом перетворення аудіо в текст.

Загалом, розробка голосового помічника призначена для опрацювання певних завдань і предстала, як зручний спосіб взаємодії з компонентами окремого додатку чи самого пристрою Android без допомоги рук.

Аналіз досліджень. Було оглянуто статті, навчальні матеріали, онлайн-форуми, інтернет-джерела [1-4], відео матеріали [5-6] та з'ясовано, що усі вище розглянуті попередні дослідження акцентують увагу на ключових функціональностях голосових помічників для Android, таких як розпізнавання мовлення, взаємодія з іншими додатками та компонентами смартфона, а також реалізація мовлення в текст.

Проаналізована оцінка ефективності реалізації функцій у різних середовищах та виявлено, що інтеграція голосового помічника з існуючими додатками та сервісами є ключовою для покращення користувацького досвіду та функціональності. При використанні даного підходу проектування була вирішена проблема складності взаємодії користувача з додатком через голосові команди, оскільки основні механіки впроваджені з урахуванням простоти використання. Також, результати дослідження допомогли у виборі середовища та мови програмування, для розробки простого голосового помічника.

У дослідженні [7] розглянуто можливості перетворення аудіо інформації в текст через хмарний сервіс Google Speech Recognition API, що дозволяє на базі цього перетворення створити простий голосовий помічник. Також, ця публікація підкреслює гнучкість та адаптивність голосових помічників до різних потреб користувачів.

Таким чином, проаналізовані джерела підтверджують доцільність використання сучасних технологій розпізнавання мовлення та обробки природної мови для створення простого голосового помічника у додатку для платформи Android. Важливими аспектами при розробці такого помічника є інтеграція з іншими сервісами, налаштування користувацьких команд та забезпечення високої точності розпізнавання мовлення.

Мета роботи полягає у наданні покрокового керівництва зі створення базового голосового помічника для Android-додатків, зосередившись на розробці простого голосового помічника під назвою «ExpertAvon».

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Розробка голосового помічника для платформи Android, зокрема «ExpertAvon», включає кілька ключових етапів, кожен з яких вимагає ретельного підходу та використання сучасних технологій. Процес розробки охоплює інтеграцію голосового розпізнавання, перетворення аудіо в текст і реалізацію відповідних реакцій додатка на отримані голосові команди. Алгоритм роботи голосового помічника можна переглянути на рисунку 1.

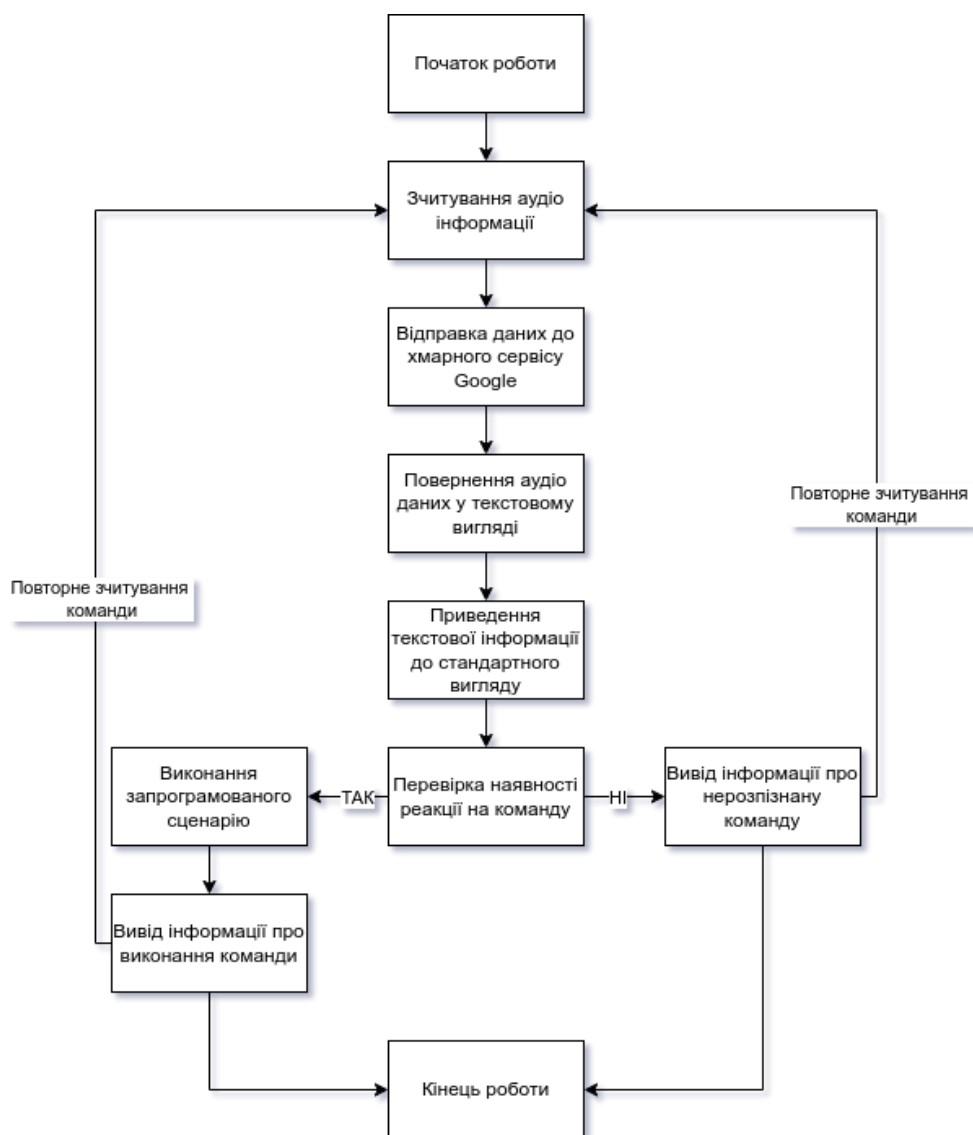


Рис. 1. Алгоритм роботи голосового помічника

На першому етапі відбувається налаштування середовища розробки. Використовується Android Studio та мова програмування Java, що забезпечує високу сумісність з платформою Android. В незалежності від того, чи інтегрується помічник у готовий додаток, чи реалізовується як окрема програма, алгоритм розробки буде однаковим. Для реалізації функціоналу розпізнавання мовлення інтегрується хмарний сервіс Google Speech Recognition API, який надає можливості точного перетворення аудіо в текст.

Наступним кроком є додавання бібліотек, необхідних для роботи з Google Speech Recognition API. Однією з таких бібліотек є бібліотека «speech.RecognizerIntent». Також, було додано декілька інших базових бібліотек. Використання цих бібліотек дозволяє додатку отримувати доступ до мікрофона пристрою, записувати аудіо та надсилати його до сервісу для розпізнавання. Важливим аспектом є налаштування дозволів та обробка можливих виключень, що може виникнути під час роботи з аудіо.

Далі йде етап розпізнавання мовлення та обробки тексту. Код який відповідає за таку обробку представлено на лістингу 1. Аудіо дані, отримані від користувача, передаються на сервер Google для обробки. Після розпізнавання, текстові дані повертаються додатку, де здійснюється їх подальша обробка. Основним завданням на цьому етапі є забезпечення високої точності розпізнавання та мінімізації помилок. Для цього можуть використовуватися різні алгоритми попередньої обробки аудіо, такі як фільтрація шуму та нормалізація гучності, однак для простого голосового помічника вони не потрібні.

Лістинг 1 - Код для налаштування зчитування голосових команд

```
213 private void startVoiceRecognitionActivity() {
214     Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
215     intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, value: "uk-UA");
216     intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
217     intent.putExtra(RecognizerIntent.EXTRA_PROMPT, (CharSequence) null);
218     startActivityForResult(intent, VOICE_RECOGNITION_REQUEST_CODE);
219 }
```

Кінець лістингу 1

На завершальному етапі реалізується взаємодія з іншими компонентами додатка або смартфона. Отриманий текст використовується для виконання відповідних команд, наприклад, запуск інших додатків, виконання пошукових запитів або управління функціями смартфона. Цей етап вимагає ретельного проектування та тестування, щоб забезпечити коректну роботу всіх функцій. Приклад коду, який відповідає за реакції на голосові команди можна переглянути на лістингу 2. Цей приклад було взято з додатку «Emergency App», у який інтегровано голосовий помічник «ExpertAvon».

Лістинг 2 - Приклад коду, який відповідає за реакцію на команду

```
221 @Override
222 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
223     super.onActivityResult(requestCode, resultCode, data);
224
225     if (requestCode == VOICE_RECOGNITION_REQUEST_CODE && resultCode == RESULT_OK) {
226         ArrayList<String> matches = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
227
228         if (matches.contains("включи телевізор")) {
229             TVImageOn.setVisibility(View.VISIBLE);
230             Toast.makeText(context: this, text: "Телевізор увімкнено", Toast.LENGTH_SHORT).show();
231             mediaPlayerNewsTV.setLooping(true);
232             mediaPlayerNewsTV.start();
233         } else {
234             Toast.makeText(context: this, text: "Невідома команда", Toast.LENGTH_SHORT).show();
235         }
236     }
237 }
238 }
```

Кінець лістингу 2

Розглянемо аспекти створення ефективного голосового помічника. Графічний інтерфейс додатку має бути інтуїтивно зрозумілим та легким у використанні. Визначення основних характеристик і функцій голосового помічника є важливим етапом перед початком розробки.

Голосовий помічник «ExpertAvon» включає в себе дві кнопки: кнопка початку зчитування команд та кнопка інформаційної довідки, яка виводить діалогове вікно з переліком доступних голосових команд. Кнопка початку роботи голосового помічника має код показаний на лістингу 3. Цей код запускає активність голосового помічника при натисненні

Лістинг 3 - Код для кнопки запуску голосового помічника

```
132 microphoneButton.setOnClickListener(new View.OnClickListener() {  
133     @Override  
134     public void onClick(View v) { startVoiceRecognitionActivity(); }  
137 });
```

Кінець лістингу 3

Особливості платформи Android дозволяють реалізувати високоякісну інтеграцію голосових команд з різними додатками та сервісами. Важливими характеристиками такого підходу є можливість адаптації до різних потреб користувачів, що забезпечує більш широкий спектр застосування голосового помічника. Інтеграція голосового помічника з іншими сервісами дозволяє значно покращити користувацький досвід та функціональність додатку.

Таким чином, розробка простого голосового помічника для платформи Android потребує системного підходу та використання сучасних технологій розпізнавання мовлення. Важливими аспектами є інтеграція з хмарними сервісами, налаштування користувацьких команд та забезпечення високої точності розпізнавання мовлення. Результати дослідження підтверджують ефективність запропонованих підходів та демонструють можливості створення зручного та функціонального голосового помічника.

Висновки.

У статті було розглянуто процес розробки простого голосового помічника для Android-додатків, названого «ExpertAvon». Основними етапами розробки стали налаштування середовища розробки, інтеграція технології розпізнавання мовлення, обробка аудіо даних та реалізація відповідних реакцій на голосові команди. Використання Google Speech Recognition API дозволило забезпечити високу точність перетворення аудіо в текст.

Дослідження підтвердило доцільність застосування сучасних технологій розпізнавання мовлення для створення простих голосових помічників. Інтеграція голосового помічника з іншими додатками та сервісами значно покращує користувацький досвід і функціональність додатків. Основними результатами роботи є створення покрокового керівництва з розробки голосового помічника, а також забезпечення його ефективної взаємодії з компонентами додатка та смартфона.

Таким чином, розробка простого голосового помічника для платформи Android, зокрема «ExpertAvon», демонструє можливості створення зручних та функціональних інструментів для взаємодії з мобільними додатками. Результати дослідження можуть бути корисними для розробників, що прагнуть інтегрувати голосові команди у свої Android-додатки.

Список бібліографічного опису

1. Голосові помічники: що це і навіщо вони HR [Ok Google, Alexa, Siri, Cortana]. HURMA. URL: <https://hurma.work/blog/voice-assistants-shho-cze-i-navishho-voni-hr-ok-google-alexa-siri-cortana/> (дата звернення: 16.05.2024).
2. Добірка найзручніших голосових помічників – bit.ua. bit.ua Медіа про життя і технології в ньому. URL: <https://bit.ua/2021/07/dlya-sproshhennya-roboty-dobirka-najzruchnishyh-golosovyh-pomichnykiv/> (дата звернення: 16.05.2024).
3. Building a voice assistant for an android java or kotlin app. Alan docs. Alan AI. URL: <https://alan.app/docs/tutorials/android/integrating-java-kotlin/> (дата звернення: 16.05.2024).
4. How to create virtual assistant apps like siri and google assistant. Cleveroad Inc. - Web and App development company. URL: <https://www.cleveroad.com/blog/how-to-create-virtual-assistant-apps-like-siri-and-google-assistant/> (дата звернення: 17.05.2024).
5. GoXR3Plus Studio. Java speech recognition tutorial (part 1), 2016. YouTube. URL: <https://www.youtube.com/watch?v=R8vsXKFTee0> (дата звернення: 17.05.2024).

6. Infybuzz. Create your own voice assistant in java. Just like google assistant, siri OR amazon alexa, 2021. YouTube. URL: <https://www.youtube.com/watch?v=6Ip42tbq4DM> (дата звернення: 17.05.2024).
7. How to convert speech to text in android? - geeksforgeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/how-to-convert-speech-to-text-in-android/> (дата звернення: 17.05.2024).

References

1. Voice assistants: what they are and why HR needs them [Ok Google, Alexa, Siri, Cortana]. HURMA. URL: <https://hurma.work/blog/voice-assistants-shho-cze-i-navishho-voni-hr-ok-google-alexa-siri-cortana/> (accessed May 16, 2024).
2. A selection of the most convenient voice assistants - bit.ua. bit.ua Media about life and technology in it. URL: <https://bit.ua/2021/07/dlya-sproshhennya-roboty-dobirka-najzruchnishyh-golosovyh-pomichnykiv/> (accessed on May 16, 2024).
3. Building a voice assistant for an android java or kotlin app. Alan docs. Alan AI. URL: <https://alan.app/docs/tutorials/android/integrating-java-kotlin/> (accessed May 16, 2024).
4. How to create virtual assistant apps like siri and google assistant. Cleveroad Inc. is a Web and App development company. URL: <https://www.cleveroad.com/blog/how-to-create-virtual-assistant-apps-like-siri-and-google-assistant/> (accessed on May 17, 2024).
5. GoXR3Plus Studio. Java speech recognition tutorial (part 1), 2016. YouTube. URL: <https://www.youtube.com/watch?v=R8vsXKFTee0> (accessed May 17, 2024).
6. Infybuzz. Create your own voice assistant in java. Just like google assistant, siri OR amazon alexa, 2021. YouTube. URL: <https://www.youtube.com/watch?v=6Ip42tbq4DM> (accessed 17.05.2024).
7. How to convert speech to text in android? - geeksforgeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/how-to-convert-speech-to-text-in-android/> (accessed May 17, 2024).