

Mobile Apps Development: Towards Incorporating Regional and National languages in Flutter App

Kamran Khan¹, Dr. Isma Farah Siddiqui¹, Salahuddin Saddar¹, Hina Hussain¹, Umair Ayaz¹, and Sadaquat Ali Ruk²

Abstract:

As technology spreads rapidly worldwide, people from different countries have started using portable devices like phones, tablets, or laptops. With the growing trend of globalization, people worldwide are now using these portable devices to access various apps and services, which means that apps should be developed in a way that caters to the needs of these diverse users. Most apps are unavailable in multiple languages, which can be frustrating for users who are not fluent in the language in which the app is developed. Developing a multilingual app and taking input in multiple languages is still a headache for developers. This research aims to develop a multilingual keyboard package for Flutter apps. This package allows developers to add a multilingual keyboard to their apps easily and quickly. The package at this stage only supports English, Urdu, and Sindhi languages.

The proposed approach also added some special characters, a numeric keyboard, and language keyboards. The keyboard interface is highly customizable, developers can customize it as per their preferences, and they can change different properties of keyboards, including key text color, key background color, keyboard background color, and shadows. This package allows users to enter text in multiple languages in the same or different input fields. After the publication of this package, the authors conducted a survey in which questions like the performance, accuracy, and interface of the keyboards were asked of the developers. In the survey, researchers asked for the developer's suggestions and feedback about keyboards and other languages they want us to add to the package.

Keywords: *multilingual keyboards, flutter package, in-app keyboards.*

1. Introduction

We live in an era where technology is becoming more and more powerful and compact. People love to use portable devices to carry them easily wherever they go. Portable devices allow users to stay connected with

their family, friends, colleagues, clients, bosses, and team leads.

Mobile devices have become necessary for everyone, and mobile technology is evolving quickly. Mobile traffic has increased than desktop traffic after November 2019 [1]. As the usage of mobile devices has increased, new companies and technologies are coming into being. Android, iOS, Windows, and

¹Department of Software Engineering, Mehran University of Engineering & Technology, Jamshoro, Pakistan.

²Department of Computer Science, Shah Abdul Latif University, Ghotki, Pakistan.

Corresponding Author: isma.farah@faculty.muet.edu.pk

BlackBerry are operating systems currently used in mobile devices. Different development technologies or languages are used to develop an application for these OS. Developers must maintain a separate code base for each app which is a lengthy and costly process. To reduce the time and cost of the development process, developers prefer to choose cross-platform technologies. Cross-Platform App development is a way to develop apps for multiple operating systems with a single code base.

Flutter is a framework used to develop cross-platform apps which will run on both Android & iOS. Flutter reduces the complexity, LOC, and development time to approximately 50 percent [2]. Other cross-platform technologies use a bridge to communicate with the native language, but Flutter removes this bridge and runs the app in an environment mostly built on C++ [3]. Unlike other cross-platform technologies, Flutter does not convert code to a native language and does not use a web wrapper to run code. Flutter uses its engine increasing performance and load UI components efficiently [4].

Since Flutter is new technology evolving daily, so is its community. Flutter contributors are publishing libraries rapidly on <https://pub.dev/>. <https://pub.dev/> is the platform where flutter libraries are published. Besides publishing the library on <https://pub.dev/>, developers create a public repository on GitHub to push the library's code. Developers can read all the library code and open an issue on GitHub if they find any. Developers can also contribute to the library by GitHub pull requests. Flutter 2 was released on 3rd March 2021, with the support of Null-Safety [4], which was very rare in other mobile application development technologies [5]. Null-Safety means the app will not encounter a Null-Pointer exception at runtime. Flutter 2 and onwards, all the versions of Flutter support Null-Safety and old projects should be migrated to Null-Safety to debug and deploy apps.

2. Literature Review

Studies on text input methods for mobile devices started with the invention of smartphones. Many researchers have researched keyboard types, alphabet placement, and multilingual keyboards.

Most students use keyboards for the WhatsApp application, concluded by Mustafa Serkan Abdusselam through a survey in which 238 candidates participated [6]. A survey concluded that the study on keyboards should be done on the end-user side, where users use keyboards with their personalization [7]. Phone manufacturers have done a comparative study on keyboards, and they discussed demanded features, lacking areas of keyboards, and how typing speed can be increased [8]. The Octopus keyboard is the fastest, producing fewer errors than T+ and Curve keyboards [9].

3. Overview Of Selected Technology

Flutter is a UI toolkit developed by Google [15]. Flutter is not a programming language. Instead, it is an SDK with prewritten code that includes ready-to-use and customizable widgets [12]. Flutter SDK is built on Dart programming language; in Flutter, every application is written using Dart programming language.

3.1. Flutter

Flutter is an open-source, cross-platform framework for developing high-quality, high-performance apps for various operating systems, including Android, iOS, Windows, Linux, Mac, and Google Fuchsia, from a single codebase [10]. Google publicly released Flutter in May 2017. Google has chosen Flutter as the application-level framework for its next-generation operating system. Flutter is unique because it relies on the device's OEM widgets rather than consuming web views [11]. Flutter renders each view component using its high-performance rendering engine. Flutter's engine, primarily written in C++, supports

low-level rendering via Google's Skia graphics library or the custom "Impeller" graphics layer.

3.1.1 Features of Flutter

- Flutter's hot reload feature allows seeing changes in the code immediately reflected on the UI [15]. This speeds up the process of working on the application's UI. As a result, developers can correct errors more quickly and save costs and time.
- Using Flutter, developers can write code that works on multiple platforms at the same time [2][3][15]. It saves developers or companies time, money, and effort.
- In Flutter, every UI component is defined as a widget. Text, buttons, padding, or a menu are all examples of widgets [2][12]. Flutter can generate complex and customized widgets according to the application's requirements.
- Unlike most cross-platform frameworks, Flutter does not use intermediate code representations or interpretations. The Flutter application is written directly into machine code [1], eliminating any performance issues caused by the interpretation process. Flutter provides a fully compiled release application ahead of time.

3.2 Dart Programming Language

Dart is a general-purpose, object-oriented programming language with C-style syntax that Google created in 2011. To create an app with Flutter, developers need to code in Dart. Like other programming languages such as Java, Kotlin, Swift, and JavaScript, Dart is easy to learn. According to Google, Dart is a client-optimized language for building fast apps on any platform.

4. Methodology

This section outlines the step-by-step procedure for creating, testing, and publishing a multilingual keyboard library. It also describes gathering alphabets, keyboard design, development, testing, and publishing on <https://pub.dev>.

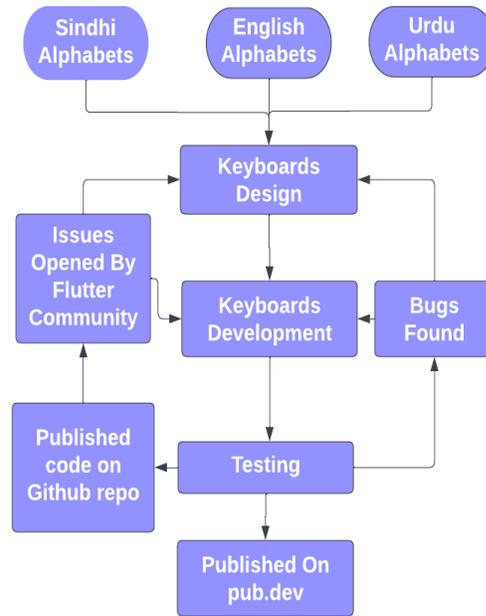


Fig. 1 Methodology Diagram

Fig.1 describes a step-by-step process to develop, debug, and publish a multilingual keyboard library to <https://pub.dev>. The process starts with gathering the alphabet of languages (Urdu, Sindhi, and English). Once the alphabet is gathered, the design phase starts. The development phase follows it, and after the development testing phase comes. Once the package is stable enough for the community to use, then the package is published to <https://pub.dev>, and simultaneously code of the package is published to the GitHub repository.

4.1 Gathering Alphabets

The authors gathered alphabets of Sindhi, Urdu, and English languages from different

research papers. All the alphabets are gathered from valid sources. The Urdu language consists of 37 fundamentals alphabets [13], and there is some character variation that is made by combining two or more alphabets, such as "کھ" which is the combination of 2 Urdu alphabets, "ک" and "ھ." The Sindhi language consists of 52 alphabets [14], and The English language has 26 capital alphabets and 26 small alphabets. All numbers from 0 to 9 have been taken for the numeric keyboard. Moreover, for the special character keyboard, familiar special characters have been chosen for our daily lives.

4.2 Keyboards Design

The next step after collecting the alphabet is designing the keyboards. Two keyboards with special characters and numeric keys have been created, two for each language. We created two keyboards with 26 alphabets each because the Sindhi language has 52 different letters. In the case of Urdu, which uses a script with 37 alphabets, developers may need to create specialized keyboards to ensure that users can type in the language effectively. To accommodate the 37 alphabets, two keyboards were designed for Urdu., one with 26 alphabets and the other with 11 remaining alphabets and some with a variation. Two keyboards are designed for the English language, one for upper-case alphabets and the other for lower-case alphabets.

4.3 Keyboards Development

After finishing the Design, all the keyboards were developed and incorporated fundamental features of keyboards, such as adding space, new lines, jumping to the next field, and the done action. At the time of development, researchers focused on features and the UI of keyboards. The package is developed in such a way that developers can easily personalize keyboard UI in their apps. The developers can choose the UI of the keyboards of their choice. The keyboard

background color, key background color, key text color, key spacing, key shadows, and border-radius can be changed easily to achieve the desired UI.

4.4 Testing

Once keyboards were developed testing phase was initiated. The keyboard design, accuracy, performance, and efficiency were tested in the testing phase. Design issues were found, and the design phase was reinitiated to resolve those design issues. Similarly, accuracy, performance, and efficiency issues were found and resolved in the development phase.

4.5 Publishing Package on the <https://pub.dev>

After completing the testing phase, when the package was ready for the community to use, the package was published on <https://pub.dev>¹. The package's name is "Artistic Multilingual Keyboards" and is available at <https://pub.dev>².

4.6 Publishing source code on GitHub

The source code of the package is available on the GitHub repository³. It is available for the community to contribute to the package. Following are the ways the community can contribute to the package:

- Test and report if any issue is found.
- Resolve reported issues.
- Add a new feature to the package.
- Suggest some good changes.

Anybody can report an issue found on the package by simply opening an issue on the GitHub repository and describing all the steps to reproduce the issue. One can add a new feature in the package by creating a new branch from the main and its pull request on the GitHub repository. The code owner then reviews that pull request and merges the code into the main code.

¹ <https://pub.dev>

² https://pub.dev/packages/artistic_multilingual_keyboards

³ https://github.com/kamran8545/artistic_multilingual_keyboards

5. Tools & Technologies

- **Android Studio** is an IDE used to develop mobile applications [16].
- **Flutter SDK** is a set of libraries containing the source code of the Flutter framework [17].
- **Dart SDK** is a set of libraries containing the source code of the Dart programming language on which the Flutter framework is built [18].
- **Flutter & Dart Plugins** enable developers to develop Flutter apps on Android Studio [19].
- **A mobile device** for testing purposes could be a physical device or an emulator.
- **Git** is a tool to manage source code on the GitHub repository or any other platform [20].

6. Results & Discussion

This research aims to develop a multilingual Flutter package allowing developers to add multilingual keyboards to their applications. The proposed approach has chosen three languages Sindhi, Urdu, and English. Along with these three languages keyboards a numeric keyboard and some special characters keyboards also been developed. By using this package developers can easily add in-app language specific keyboard in their apps.

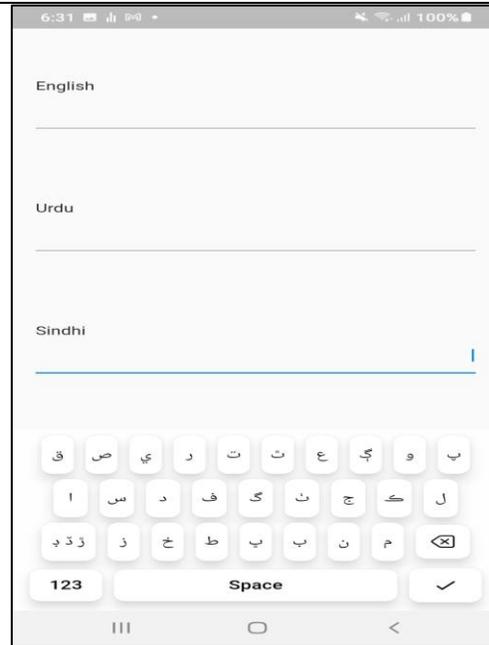


Fig. 2 Sindhi Keyboard of 26 alphabets

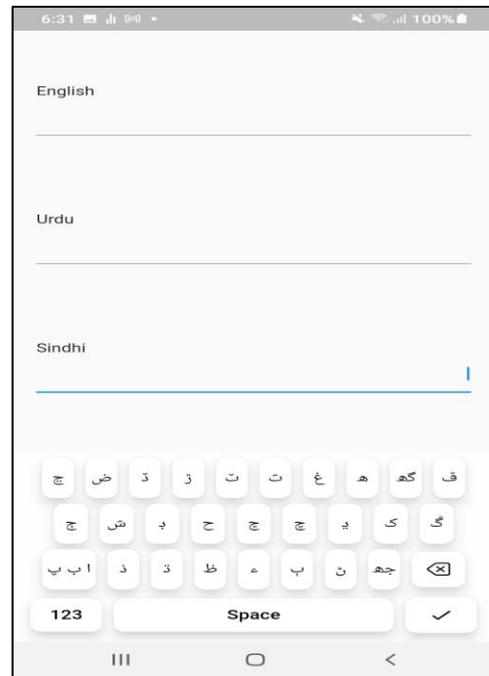


Fig. 3 Sindhi Keyboard of remaining 26 alphabets

Figures 2 and 3 show the Sindhi language keyboards containing 26 alphabets each.

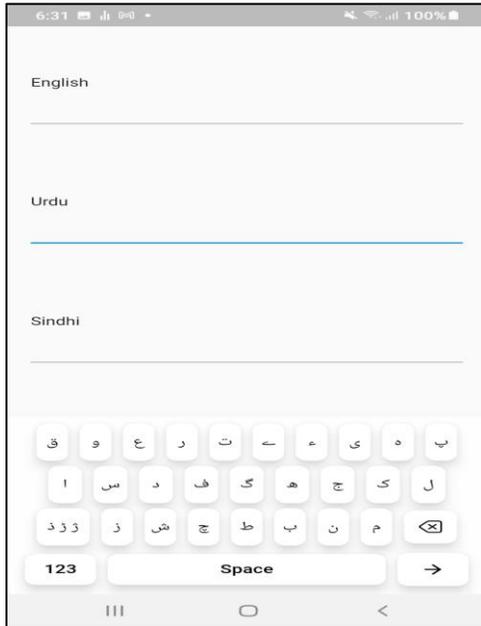


Fig. 4 Urdu Keyboard of 26 Alphabets

Fig. 4 shows the Urdu language containing 26 alphabets, and Fig. 5 shows the Urdu language keyboards containing the remaining 11 alphabets and some alphabet variations.

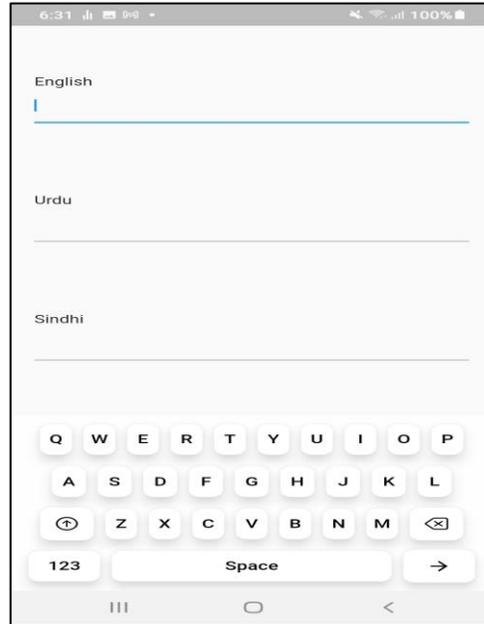


Fig. 6 English Uppercase Keyboard

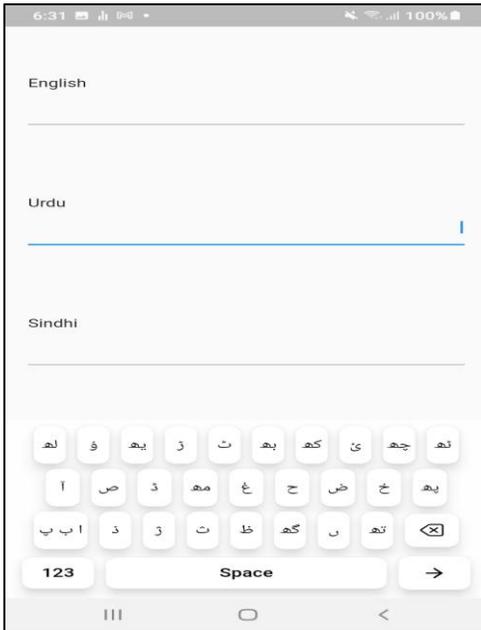


Fig. 5 Urdu Keyboard of remaining 11 alphabets and with some variations

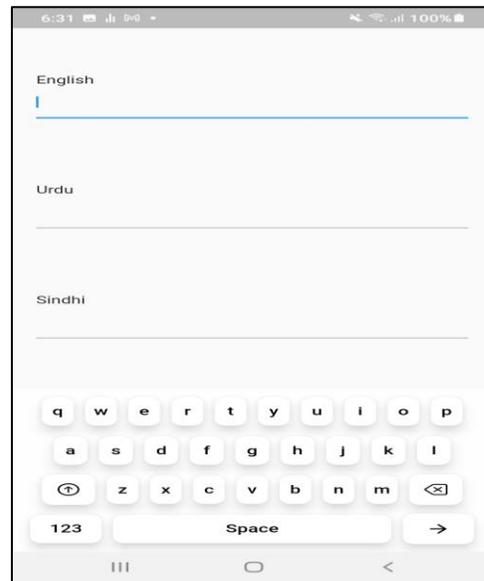


Fig. 7 English Keyboard

Figures 6 and 7 show the English language keyboards containing English Uppercase and lower-case alphabets, respectively.

6.1 Customized UI

Following are some customized designs of keyboards. By changing some properties of keyboards, desired Design can be made. Fig. 8 and Fig. 9 show some UI customization, but it is not the only UI developers can use in their apps. UI of keyboards is highly customizable; they can change the keyboard background color, key background color, key text color, key shadow, key corners radius, and other properties to achieve their desired UI.

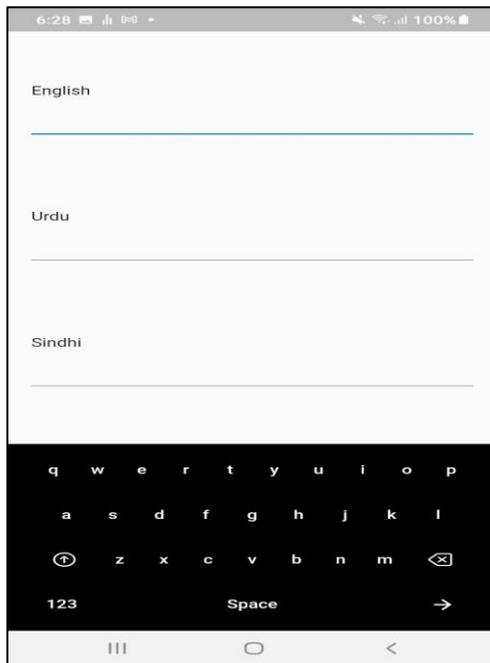


Fig. 8 Customized UI

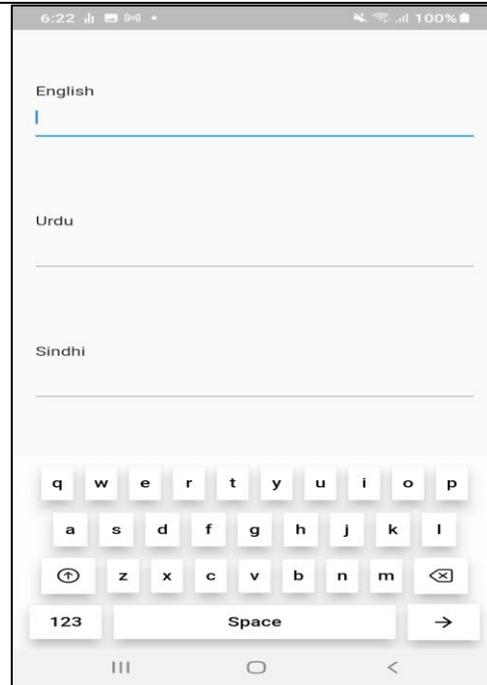


Fig. 9 Customized UI

7. Performance Evaluation

After the package's release, the researchers polled the Flutter community. The survey's participants ranged in experience from beginners to seniors. 109 people in total took part in the survey. Participants were questioned in the survey regarding the keyboard's UI, performance, accuracy, efficiency, etc.

Following are the few questions asked by developers:

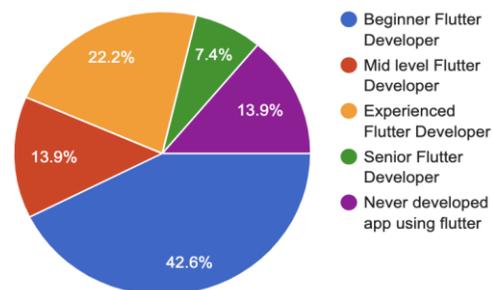


Fig. 10 Participants' experience level.

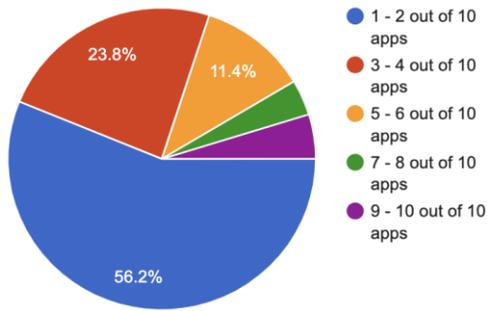


Fig. 11 How frequently do participants use or see multilingual keyboards in mobile apps?

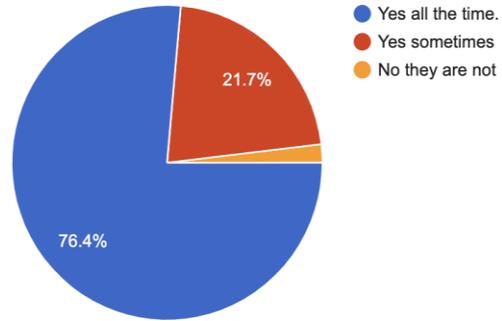


Fig. 14 Are the keyboards typing alphabets with accuracy?

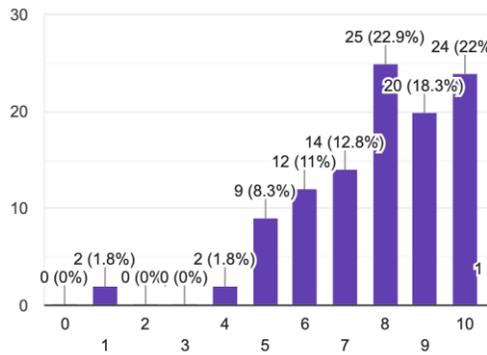


Fig. 12 Rate the keyboard's UI from 0 to 10.

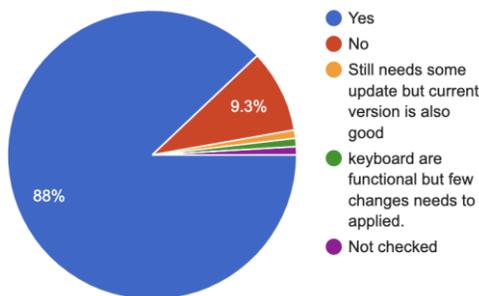


Fig. 13 Are all the keyboards functioning correctly?

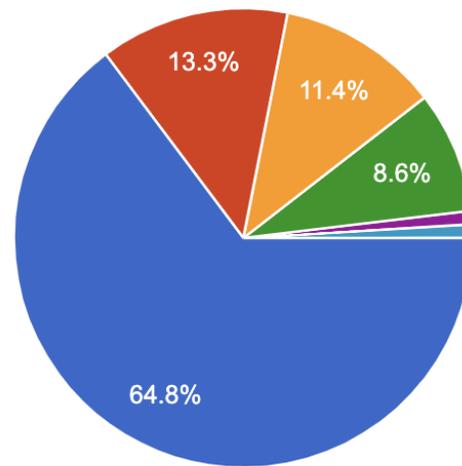


Fig. 15 Are all the keyboards working fine when you are typing more rapidly?

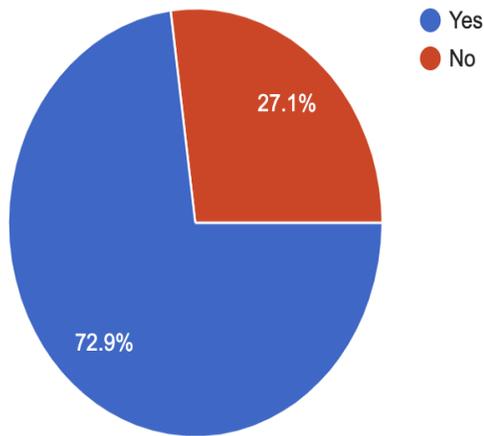


Fig. 16 Should we add more languages to this package?

S#	Language	No of suggestions
1	French	6
2	Hindi	6
3	Arabic	8
4	Spanish	7
5	Chinese	4
6	Japanese	3
7	Pashto	4
8	German	3
9	Russian	1
10	Korean	3
11	Punjabi	3
12	Balochi	4
13	Saraiki	1
14	Brohi	1
15	Xhosa	2

Table. 1 Suggested Languages

Table.1 shows the language name, and no of times a language has been suggested to add to the package.

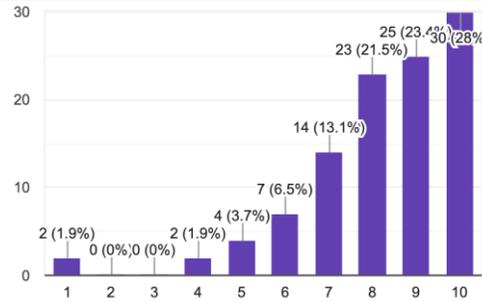


Fig. 17 Rate this package from 0 to 10.

8. Conclusion & Future Work

In this research, an open-source Flutter package has been developed and published to add multilingual support to Flutter apps. By adding this package to their apps, developers can take input from users in multiple languages. They can take multilingual input in a single field or multiple fields. This package will reduce the efforts of end-users. They do not have to download multiple languages to their devices to input in multiple languages. The researchers also surveyed the developers to improve the package's performance, accuracy, and efficiency; moreover, they were asked about other languages developers want us to add to the package. This research further aims to add more languages to the package and improve its performance, accuracy, and efficiency by evaluating the conducted survey.

REFERENCES

- [1] Tahsildar, Aakanksha, et al. "Application development using flutter." *International Research Journal of Modernization in Engineering Technology and Science* 2.08 (2020): 1262-1266.
- [2] Olsson, Matilda. "A Comparison of Performance and Looks Between Flutter and Native Applications: When to prefer Flutter over native in mobile application development." (2020).
- [3] Ameen, Siddeeq Y., and Dilkhaz Y. Mohammed. "Developing Cross-Platform Library Using Flutter." *European Journal*

of Engineering and Technology Research 7.2 (2022): 18-21.

[4] "FAQ," Flutter. [Online]. Available: <https://docs.flutter.dev/resources/faq>.

[5] Publication, I. J. R. A. S. E. T. (2021, 5th June). A research paper on a progress tracking application using Flutter and Firebase. International Journal for Research in Applied Science & Engineering Technology. Retrieved 8th March 2022, from https://www.academia.edu/49135635/A_Research_Paper_on_a_Progress_Tracking_Application_using_Flutter_and_Firebase

[6] Abdusselam, Mustafa Serkan. "A Research on Students' Preferences for Mobile On-Screen Keyboard." Shanlax International Journal of Education 9.2 (2021): 46-53.

[7] Buschek, Daniel, Benjamin Bisinger, and Florian Alt. "ResearchIME: A mobile keyboard application for studying free typing behavior in the wild." Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. 2018.

[8] Jog, Yatin, and Veronica Singh. "Smart Keyboards: Need of Customization and Personalization of Mobile Keyboards." Int. Journal of Latest Technology in Engineering, Management & Applied Science 3.9 (2014): 115-119.

[9] Cuaresma, Justin, and I. Scott MacKenzie. "A study of variations of Qwerty soft keyboards for mobile phones." Proceedings of the International Conference on Multimedia and Human-Computer Interaction-MHCI. 2013.

[10] Hina Hussain, Kamran Khan, Faiza Farooqui, Dr. Qasim Ali Arain, Dr. Isma Farah Siddiqui. "Comparative Study of Android Native and Flutter App Development." KSII The 13th International Conference on Internet (ICONI) At Jeju, South Korea, 2021. 99-102.

[11] Aakanksha Tashildar, Nisha Shah, Rushabh Gala, Trishul Giri and Pranali Chavhan. "APPLICATION DEVELOPMENT USING FLUTTER." International Research Journal of Modernization in Engineering Technology and Science (2020): 1262-1266.

[12] Shreya A. Bhagat, Sakshi G. Dudhalkar, Prathmesh D. Kelapure, Aniket S. Kokare, Prof. Sudesh A. Bachwani. "Review on Mobile Application Development Based." International Journal for Research in Applied Science & Engineering Technology (2022): 803-809.

[13] Khan, M. Aamir, M. Abid Khan, and M. Naveed Ali. "Design of Urdu virtual keyboard." *Proc. Conf on Language & Technology*. 2009.

[14] Sodhar, Irum Naz, Akhtar Hussain Jalbani, and Muhammad Ibrahim Channa. "Identification of issues and challenges in romanized Sindhi text." *Editorial Preface From the Desk of Managing Editor...* 10.9 (2019).

[15] Dagne, Lukas. "Flutter for cross-platform App and SDK development." (2019).

[16] Download Android Studio & App Tools. Android Developers. (n.d.). Retrieved 30th January 2023, from <https://developer.android.com/studio>

[17] Flutter. (n.d.). Retrieved 30th January 2023, from <https://flutter.dev/>

[18] Dart programming language. Dart. (n.d.). Retrieved 30th January 2023, from <https://dart.dev/>

[19] Flutter & Dart Plugins. Flutter. (n.d.). Retrieved 30th January 2023, from <https://docs.flutter.dev/get-started/editor?tab=androidstudio>

[20] Git. (n.d.). Retrieved 30th January 2023, from <https://git-scm.com/>