Review the Finished Mobile App

Why the app is suitable for the user requirements and target audience.

Our target audience is kids, so we have to make everything very primitive and easy-to-use.

So we made a game-coloured quiz, where users can play the pronunciation of letters right during a quiz!

What did we also do for user requirements?

* Cross-platform application - because of the fact that everyone may have different devices with different architecture, UI guidelines, SDK, so now almost everyone is able to use our application via mobile devices(such a iOS or Android device), standalone Windows/MacOS/Linux, and a Web version for everyone else! We did via Flutter&Dart as a main development tool.
* Adaptive & simple UI - because again our target audience is kids, adaptive because users have different devices with different resolution, display quality.
* Offline mode - meaning users can learn and play quiz without internet connection or any authorisation
* Demonstrating material via illustrations and sounds to get more attention and hold users
* Pronunciation learning and practice - game-quiz mode are very suitable for that.
* And the main requirements from users - learning letters, you can learn letters with our app, and after that enjoy gamemode!

Why the app is suitable for the intended purpose of the app.

Small kids have to be able to learn the alphabet without any problems, our app is intuitive and understandable, if not - they may ask parents to help.

We have funny icons and illustrations for letters/buttons/pages, everything with minimal-coloured style to attract kids attention, we have pronunciation players.

So like that we give all main features which the user wants, and also try to hold user attention for much longer as possible.

Consider how constraints, user feedback and testing have affected the

suitability of the app.

We could add more interactive features to the quiz-game but young kids can’t get so much information at the same time, they need to just learn or just play, so we decided to not add hard features to the app as it will cause problems with usability as this is a limitation.

After good feedback we understood that we were on the right track, and tried to improve performance, and small features like icon colours , button/font size.

While development we understood that some features work incorrect, like a after finishing quiz you could miss one question, as a builder had incorrect rules how he need to do, so we consider to change that rules, and rewrite quiz-builder, that’s made possible better performance. Also we had a bug when after authorisation users could catch a segmentation fault as http-feature was too long, and auth controller couldn't work with that data, so we again made some changes to improve the app and make it faster. Same with UI/UX, only after a few games, and a few minutes you could understand that some of the buttons had a different size, and bad positioning, so after a few updates - the app looked better.

Evaluate your initial designs and the completed app, justifying any changes made.

Icons and images, because ours was in really bad quality, and not suitable for use in production mode, so we made such a more minimal design to make it look more professional and simpler.

Make at least three specific suggestions for improvement for the completed program to ensure it is fully functional, well-coded and fit for the intended purpose.

* Unit and open testing - to find bugs
* Add server-side calculations - it’s faster, as safer
* Add colour-blind and other issues management, and theme changer - as not all can use default colours