REPORT for developing A cross-platform SUDOKU APP

# Introduction

The aim of this report is to evaluate the potential of various cross-platform development solutions for creating a Sudoku app. This report covers the differences between popular cross-platform solutions as well as the pros and cons of each solution. Then lastly, the rationale behind the chosen solution will be explained in detail.

# Analysis of cross-platform technologies

This section of the report focuses on discussing the appeals of the different cross-platform solutions along with their potential hindrances.

## Ionic

Ionic is a popular framework used for developing hybrid mobile applications (web applications packaged inside a native wrapper).

One of the biggest appeals for using ionic is the ability to have a single codebase as opposed to maintaining separate ones for IOS, Android and other platforms. This means that the features and UI remain consistent across different platforms.

Another reason for its popularity today can be linked to the use of well-established web technologies which includes HTML, CSS, and JavaScript. This makes program development using ionic versatile, as it can easily utilize new web standards or capabilities. Ionic also supports integration with leading frameworks like Angular, React and Vue.

So, what are the drawbacks of using Ionic? Ionic is not truly native, since it is essentially a web app running inside a WebView, it will not perform as well as true native apps when it comes to graphics intensive applications or tasks. This also means that this extra layer between the app and the hardware can introduce some performance overhead, which could negatively impact the user experience.

## Flutter

Flutter is an open source UI toolkit created by Google that was released in 2017 for development in cross-platform applications.

Like ionic, Flutter uses a single codebase for development across devices. But unlike Ionic, Flutter uses the Dart language which compiles ARM or Intel machine code for fast performance on any device. It also makes use of its own graphics engine, Skia, to render visuals.

Flutter is also known for its rich set of highly customizable widgets for the creation of complex UIs and due to its own rendering engine, the UI looks consistent across different platforms.

The ecosystem and community of flutter is expected to grow. Considering Google’s backing of Flutter, its longevity and consistent update rollouts is expected.

Flutter, although rapidly growing, is still young and is newer than alternatives like React Native. This could mean fewer third-party libraries in some specific niche use cases. Dart is also not as widely adopted as other big languages like JavaScript and is single threaded (multithreading workarounds aren’t as straight forward as native multithreading).

## React Native

React Native is an opensource UI framework developed by Meta Platforms which was released in 2015. It uses the React framework along with native platform capabilities.

One of the defining features of react native is that instead of using HTML and the DOM as building blocks for the UI, it uses native components that are available for IOS and Android. React Native also supports platform specific changes where developers can customize the UI for different platforms. These two features combined, help to create a seamless native app experience.

React has a mature ecosystem with a well established community and with Facebook’s backing, continues to receive updates, improvements, and support.

React Native leverages JavaScript, one of the most widely adopted programming languages globally. This means that developers can tap into a vast reservoir of resources, libraries, and tools available for JavaScript.

Even though React Native’s performance is generally good for most cases, CPU-intensive tasks and complex animations could potentially introduce performance bottlenecks, and in these cases native code might be more efficient.

Unfortunately, not everything can be achieved with React Native directly. Some functionalities, especially those accessing specific hardware or OS features, might require developers to create native modules which could complicate the development process.

Integrating React Native could prove difficult when trying to work with native modules or when used as part of a hybrid app.

## Cordova

Cordova, previously known as PhoneGap, is a mobile application development framework developed by Nitobi that enables developers to build mobile apps using web technologies like HTML, CSS, and JavaScript. Among other frameworks and tools Ionic was built on top of Cordova to make use of Cordovas core tools.

When it comes to integration with other frameworks, Cordova can be combined with other popular frameworks such as Angular, React, or Vue.js, allowing developers to use their preferred UI libraries.

Cordova has also been around for a while and has over the years developed a large community which means more resources and third-party plugins. Since version 1.9, Cordova has made it possible to mix native and hybrid code snippets.

As for most web-based cross-platform technologies for mobile application development, Cordova applications also run slower compared to native applications with similar functionality.

## Kirigami

Kirigami is a lightweight user interface framework developed by KDE, an international free software community that develops free and open-source software, that was mainly designed to allow applications to adapt their UI based on the platform they’re executed on (which makes them usable on both desktop and mobile platforms).

QML (Qt Meta-Object language), which is a part of the Qt framework, is the language used for UI design.

Being a KDE product, Kirigami offers good integration between other KDE software and tools, contributing to its own long-standing and robust ecosystem.

Kirigami promotes a modular design, which makes it easy to reuse components across different parts of an application or projects.

Although it is properly functional and still receiving updates it has a smaller community compared to those discussed earlier. This can add additional challenges compared to other alternatives.

# **Choosing Flutter for Sudoku Game Development**

After analyzing various cross-platform technologies for developing a Sudoku game, Flutter emerged as the optimal choice for several reasons.

As mentioned before, the Skia graphics library ensures that the UI is drawn at 60fps (or 120 depending on the device). This is noticeable when implementing features like hints, errors, animations, or transitions when completing a puzzle.

Perhaps the biggest deciding factor was Flutter’s huge widget library compared to for example React Natives minimalistic approach, which encompasses only basic components (while the rest is left for the open source community to maintain, which means lots of third-party dependencies).

Like the other alternatives it supports single codebase implementation which makes it easier to maintain and update the game across different platforms.

Unlike most other alternatives that use JavaScript, Flutter’s Dart language offers strong typing. This will be beneficial for game development as catching potential bugs early could save countless hours of debugging a simple issue.

Flutter, like some of the other alternatives, also supports hot reload, which is the ability to view changes made when programming without fully restarting the application. This will save time and resources in the long run.

Though the primary focus is on mobile platforms, Flutter also has growing support for web and desktop. This could be advantageous if there’s ever a desire to expand the Sudoku game beyond mobile platforms in the future.

In conclusion, for the Sudoku game project, Flutter offers a blend of performance, rich UI capabilities, and the flexibility of deployment across platforms. Its features align well with the needs of a game that requires an engaging user interface and seamless performance. Thus, Flutter stands out as the most fitting choice for this implementation.

# Bibliography

Cordova. (n.d.). *Apache Cordova.* Retrieved from cordova.apache.org: https://cordova.apache.org/

Fireship. (2021, September 28). *React Native in 100 seconds - YouTube.* Retrieved from YouTube: https://youtu.be/gvkqT\_Uoahw?si=prmdnaL0pz8h\_7hR

Flutter. (n.d.). *Flutter - Build apps for any screen.* Retrieved from flutter.dev: https://flutter.dev/

Ionic. (n.d.). *Ionic Framework - The Cross-Platform App Development Leader.* Retrieved from ionicframwork.com: https://ionicframework.com/

Kirigami. (n.d.). *Developer.* Retrieved from develop.kde.org: https://develop.kde.org/frameworks/kirigami//