Framework solution to develop "ToDoNotes".

Since this is not a complex application, I propose this application to be developed as a Hybrid application. If it has complex features that require to access the device's full potential and capabilities then it's better to go for a native app. Even though someone has knowledge of native Android and iOS app development and money to go for Native development, it's just a waste of time and money developing a simple app such as a To-do app natively for each OS.

As the development framework for this ToDoNotes, I prefer to use the Reactnative mobile application framework since Reactnative is one of the free Cross-Platform mobile development frameworks and it's been using for a long time and it has a bigger community to support.

Compare and contrast my frameworks with other available mobile frameworks.

App Frameworks		
Hybrid	**Native**	**Web**
Pros	Pros	Pros
Lowest cost of development. As you are not going to create several versions for different app stores.	As native apps are developed specifically for a certain operating system, they are much faster.	Easier to maintain because they use one code base across multiple mobile platforms.
Easy to scale up to a variety of platforms and OSes. This is because web technology is almost 100% similar across different platforms. So, the code can be simply reused with no need to rebuild the entire app from scratch.	Native apps have the best functionality and user experience. Native apps can easily tap into any kind of device's functionality, such as camera, microphone, calendar, GPS, figure print sensors, motion sensor, etc. So, the app accessibility is very high.	You do not have to submit your app for approval to the app stores. Can be made compatible with older mobile devices and more versions.

Cons	Cons	Cons
Hybrid apps can never provide users a fully native experience. Both Android and iOS have few features exclusively for their OS. Hybrid apps never integrate with those features. Design issue remains a factor as different OSes operate differently. So, the designer needs to fix the style as per different OS.	As a native app is targeted to a specific OS, requires more time to develop for every single platform. Developers need to write specific codes for iOS, Android both. As the development cycle is longer, native apps are a little expensive compared to other hybrid or web applications. The maintenance cost is also high.	Since web apps are running in a web browser, they have a limited scope as far as accessing a device's internal features. Web apps are hard to find because they are not listed in any app stores.

App Development framework	
Reactnative	**Flutter**
Pros One codebase for two mobile platforms. Has a bigger community to support. Easy to learn and adapt	Pros One codebase for two mobile platforms User-friendly designs Less Cod
Cons Requires additional coding than the native ones Not able to integrate with some native features of the device.	Cons Not able to integrate with some native features of the device. Libraries and support are limited as compared to React Native

Value-added feature.

The user should be able to add a specific date and time for to-dos. Then the application can store it in the data-base and make a reminder about a specific to-do item on the specific day or time which was added with the to-do item to the to-do list.

Considerations that are done when deciding a mobile application framework.

- Development costs.
- Time to Market.
- Maintaining costs.
- Performance.
- Availability.
- Security.
- Hiring.
- UX.

My favorite mobile application and why.

Telegram

- It loads faster even on my old iPhone 5s compared to WhatsApp and any other messaging app.
- It has simple, user friendly and modern UI appearances.
- Small file and files that are bigger than 100mb can be transferred using Telegram without a quality reduction.