

Article

You can find more knowledge about products on this site.

- [White-label App vs. In-house App Development](#)
- [Mobile App Development Languages Used by Verestro](#)
- [Understanding White Label Applications: A Beginner's Guide](#)

White-label App vs. In-house App Development

In the world of growing IT development costs, our customers are asking more and more questions regarding [white label](#) frontend and mobile app development. In this article we will focus on the most important advantages and disadvantages of using white label frontend products.

Introduction to app development

Let's start with an introduction. As a company willing to launch new fintech products or willing to implement a mobile application for your users or employees, you need to make a business decision on how to implement it:

1. **Your own development (in-house)** - in this case you will hire developers, choose a front-end technical framework and will work with your team to implement a mobile app fully dedicated to your business.
2. **Choosing a white label product** - in this case you have to choose a vendor of a white label application, learn how to customise this application and eventually hire a team that will work on customisations necessary for your business use cases.

This choice is actually a super important decision that many business people underestimate. Let's focus on the key advantages and disadvantages of working in both models.

Scenario 1 - In-house app development

This is a common scenario for many banks, new fintechs, corporations, etc. It seems to be very easy. You will hire one or two developers for a few months and after this period of time you will have a perfect product that will include multiple functionalities. You can then resign from developers and have great business use cases for your customers. To hire developers you will usually try to hire an external IT outsourcing company that will promise to you that it is easy, fast and inexpensive.

Nothing in the above sentences is true! :) Really nothing. Many business people, especially company managers, think that front-end development is easy. That everything works great on any type of phone and that integrating backend APIs is super easy and fast.

Disadvantages

In reality, it takes time. A lot of time. To have a very good front-end product built from scratch, without bugs, you usually have to plan 12-24 months of constant development, tests, changes etc. And after this period you cannot resign from the development team. You need to have people that will work on changes, updates, will implement technical updates required by Apple, Google,

security rules etc. Let's do a quick calculation. The smallest IT team today consists of 4-5 people: backend developer, frontend developer (one or two depending on chosen technology), tester, product owner / scrum master / project manager / UX person. If you want to have fast development, this team should be bigger (8-10 people). Additionally you need hosting services - AWS or Azzure can quickly become a large part of your cost structure. You need additional software and systems connected with development work, such as Slack, Jenkins, Kubernetes, etc. All of this costs money. In short, you should expect the following costs:

- 5 people * min. 6.000 EUR average cost = 30.000 EUR monthly
- 5.000 EUR hosting monthly
- 1.000 EUR additional costs
- not including office costs, bonuses etc.

TOTAL: 36.000 EUR monthly cost -> FOR THE SMALLEST POSSIBLE TEAM!

And let's imagine that you have to spend 10 months for MVP development -> 360.0000 EUR one-time fee.

It is the cost you have to cover just to implement your MVP. Without any marketing, without any customer reactions, no sales during the period. In reality I think that you should assume that this cost of doing the implementation in such a way is 2-3 times higher than this minimum cost - almost 1 mln EUR.

Additionally, you should take into account that development done by a very small team requires technical compromises. Most likely you will not use Native iOS and Android technologies that are the best from UX perspective. IT companies will recommend to you various hybrid technologies which is always a compromise in the UX area. You will also not gather experience from other projects done in similar areas. Your mistakes will usually be first mistakes, your developer mistakes will require updates, etc.

What's even more important, you need to think about long-term development, hosting and maintenance costs. Maybe you can limit the team by 50% but costs of hosting will grow for sure with new users coming to your system. I would assume that you will have a monthly cost of 10-20.000 EUR to cover to keep the application running.

Advantages

Apologies for describing so many disadvantages but I think it is true. However, there are big advantages. If you can afford those costs and time spending, you will have full freedom. You can do with your app whatever you want to do. You can implement new features, change everything, implement new technology quickly. The dependency is only on your budget. I fully admit that this is a super important advantage that can be strategic for many start-ups and companies. I am just not sure that you must get this advantage at the very beginning of your project. Sometimes cost and time is much more important than full freedom of development. Go-to-market time may be decisive for getting new investors, growing revenues will be critical to proving that there is a problem you are solving.

Scenario 2 - White label application

Using a white label application is another strategy you can choose. In such a situation the majority of components of your application are already developed. You use an already existing product that can be customised to your requirements and you hire your developers just in case you want to make various non-standard changes in the app.

The following rules for choosing a white label application vendor are very important:

- Carefully choose **technology** - please remember that native iOS and Android solutions are just better from the UX and performance behaviour. This is what Apple and Google use for their apps.
- Check the **possibility of customisations** - make sure you understand flexibility of the product, if you can add new features, if your developers can work on the code, if you can change just colours and logo or the entire look and feel in the long run.
- Verify **experience** - check examples of other customers using this product. See how they look like, test them.
- **Prices** - obviously important. Remember to check both one-time and on-going maintenance prices. The 2nd ones are even more important.
- **Intellectual property** - very important. Is it possible that you get full IP rights to the copy of your application. Would you be able to change the development later to your own development.
- **Security and financial stability** - make sure you work with a partner that is financially stable and will not close your project in the middle of the development.

These are the most important issues that you need to check. Once you check them and they are acceptable for your business, you may get a result that your product can be 5 times faster on the market, costs can be 4 times lower, revenues will appear much faster etc. Today, the cost of white label applications can be as low as 40-60.000 EUR for development. The maintenance - 4-5.000 EUR. It can be critical for the business, especially during the first phases of growth.

Summary

I recommend that you do not believe that the world of front-end development is simple and inexpensive :) Do not make this mistake. Consider carefully if you have enough time and money. In fact, one of the most important aspects of project development is the comparison of revenues and costs. Costs are known for sure. Revenue is usually unknown. Make sure you do not overinvest. It is very easy to make a decision that you want to spend half of your 2 mln EUR on technical solution but actually it will be much, much better if you spend 200.000 EUR on a technical solution and the remaining 800.000 EUR will be used for promotions, marketing, user acquisition. This usually matters the most.

Anyhow, good luck. Thanks for reading.

Mobile App Development Languages Used by Verestro

This article summarizes programming languages & technologies we're utilizing to build our mobile solutions.

Why should I care?

If you want to use Verestro's SDKs in your mobile apps or you are interested in our Whitelabel Mobile Application, with intention of future maintenance by your own team, you may want to check if our products fit your technology stack well.

Native solution

Most of our solutions for mobile apps leverages technologies native for each mobile platform:

- For Android - Kotlin (seamlessly works also with java-based projects)
- For iOS - Swift

This enables best performance, security and integration of platform-specific features. You may also create your own native SDKs for your products that will work with our ecosystem. More details available [here](#).

Compatibility with web-based solutions

It is possible to embed widgets or entire web-based screens inside our Whitelabel Mobile Application. We'll pass you data about the currently signed in user, so you can display relevant information. More information about it is available [here](#).

Flutter, React Native and other cross-platform tools

While we don't provide SDKs for mobile apps written in Java Script & Dart, our native SDKs work great in such an environment. Both most popular cross-platform frameworks include relevant solutions for this:

- For Flutter you use [Platform Channels](#)
- For React Native you have [Native Modules](#)

Understanding White Label Applications: A Beginner's Guide

Our clients often search for effective ways to scale their offerings. The obvious way to do this is to create their own product through the entire production process, from identifying the market and its trends, to building a prototype and implementing the system, to continuous improvement. This approach requires a lot of resources and time. However, it is worth asking ourselves whether we are able to take such risks? What if the product is finished, once the demand for such solutions in the market starts to fade, or by the number of competing solutions it becomes difficult to achieve the desired results? Perhaps better results will come from using proven solutions created by experts who know the trends and have solutions in their portfolio that have paid off in the market. Due to the factors mentioned above, investors are increasingly using white label applications.

What is a White Label Application?

At its core, a **white label application** is a pre-built software product that can be rebranded and customized by businesses as their own. Think of it as a “ready-made” app that developers create, but the final product is branded with your company's logo, design, and name. The white label provider handles the coding, updates, and maintenance, while your business gets a fully functional application under your own brand. You don't have to worry about required security certifications or updates for new iOS/Android versions.

How Do White Label Applications Work?

Here's a simplified step-by-step process of how white label applications typically work:

1. **Purchase or License the Software:** Buy the application or license for a pre-developed app.
2. **Customization:** Customize the app by adding branding elements, such as logos, color schemes etc. and add your core product features through widgets or new screens.
3. **Launch and Market:** Once the app is branded, it is ready for deployment. The business can now offer it to customers as a proprietary solution.

During the customization step mentioned above there could be small and complex customizations. This is usually the critical topic how deeply you can influence white label so that it fulfills customer needs. Let's go deeper on this topic.

What is considered as custom development in the context of White Label?

Custom development involves deeper, more technical modifications to the core functionality of the white label application. This goes beyond just changing the look and feel of the app; it involves altering or adding features, adjusting workflows, or integrating the app with other systems. In

essence, custom development changes the app's functionality to meet specific business requirements.

Key Characteristics:

- **Feature Enhancements:** Adding new features or modifying existing ones to meet unique business needs (e.g., integrating custom payment gateways, modifying end-user flows, etc.).
- **Technical Adjustments:** Tweaking or rebuilding parts of the app's code to support new functionalities, or to change behaviour.
- **Custom Integrations:** Integrating the app with third-party tools like CRMs, webviews or other business platforms.
- **API Integration:** Developing custom APIs or integrating the app with existing APIs to communicate with other services.
- **Data Handling:** Modifying how the app collects, stores, or processes data.

Example: The company buys a white label app but wants to add a custom fitness tracker, integrate it with their gym's membership system, and implement push notifications. This is the custom development.

What is considered as simple customization or rebranding?

Brand change (also called **rebranding**) refers to the cosmetic adjustments made to the white label application to make it appear as though it was developed by your company. This process typically does not involve any modifications to the app's core features or functionality, but focuses instead on the visual and branding aspects.

Key Characteristics:

- **Logo Replacement:** Changing the app's logo to reflect your company's brand.
- **Color Scheme:** Customizing the app's design elements, like background colors, button styles, and fonts, to match your brand's style guide.
- **Brand Identity:** Incorporating your company's name, slogans, or taglines within the app.
- **Language changes:** Implementation of new languages in the app
- **Minor Content Updates:** Updating in-app content (like text, images, or icons) to match your brand's messaging, while the underlying functionality remains the same. That's about changing the components itself - not the position of components on the screen.

Example: A new fintech provider purchases a white label delivery app. They change the app's color scheme to match their brand, update the logo, and add their business name throughout the app. However, the functionality (eKYC, payment system, transaction history, etc.) remains unchanged. This would be a brand change.

Summary:

- **Custom Development:** Involves altering the **functionality** of the app, adding or modifying features to meet specific business requirements.

- **Brand Change:** Involves changing the **appearance** of the app by applying your company's branding elements (logo, colors, etc.) without altering its core functionality.

Save time and reduce risks by choosing Verestro's White Label solutions. With our ready-made, customizable apps, you can quickly launch your own branded product without the need for development or maintenance. Whether you need simple rebranding or more advanced custom features, Verestro's proven solutions can help you scale faster. Contact Verestro today to explore how their expertise can elevate your business.

[Wiktor Kowalczyk](#)