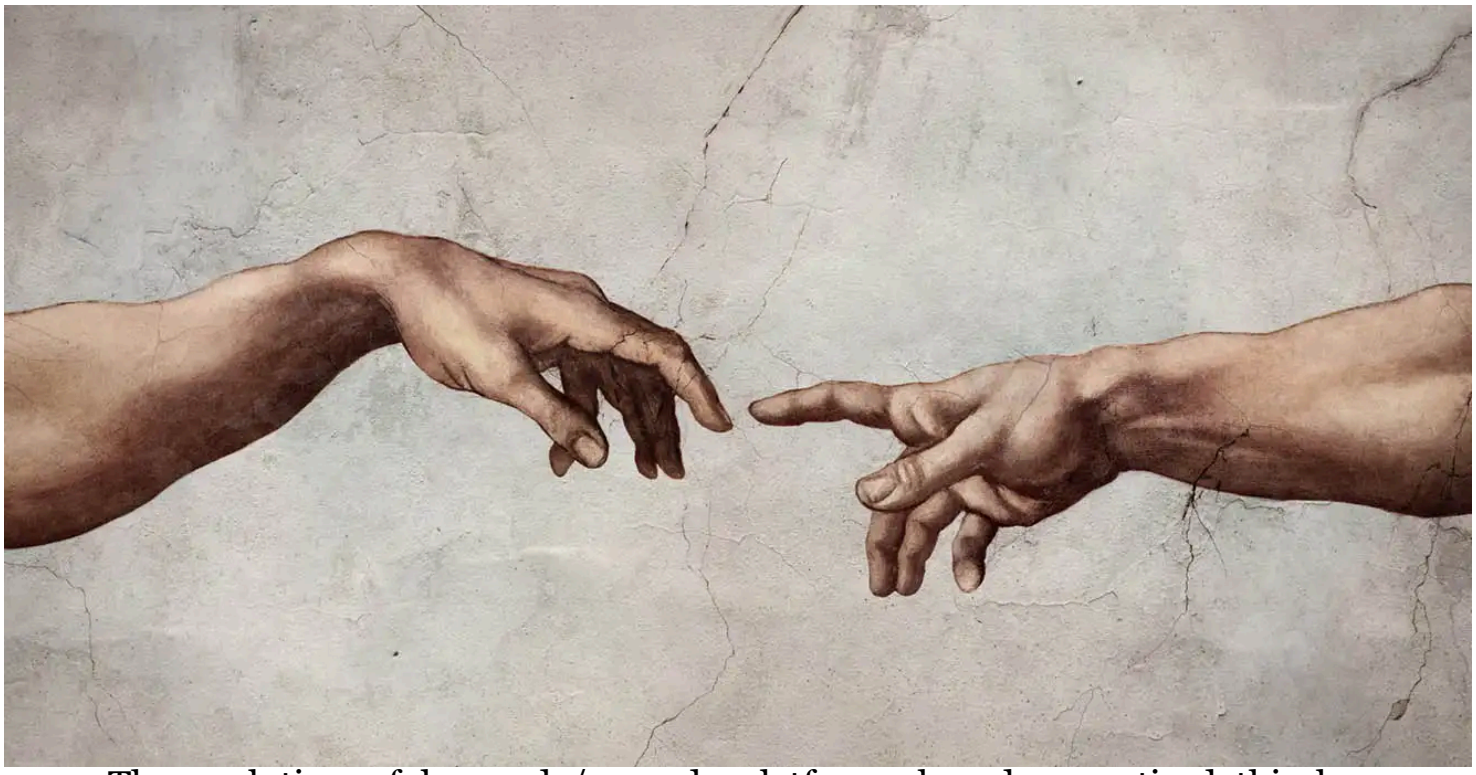


The evolution of low-code/no-code platforms has democratized this by enabling individuals with limited or no coding skills to participate in the software development process.

As a product designer, designing a developer tool can be a challenging task but getting it right is highly rewarding. It's been a year since I joined Appian Corporation and it has been a year of constant learning. Here are key things that I believe one must consider when designing a low-code/no-code developer tool:

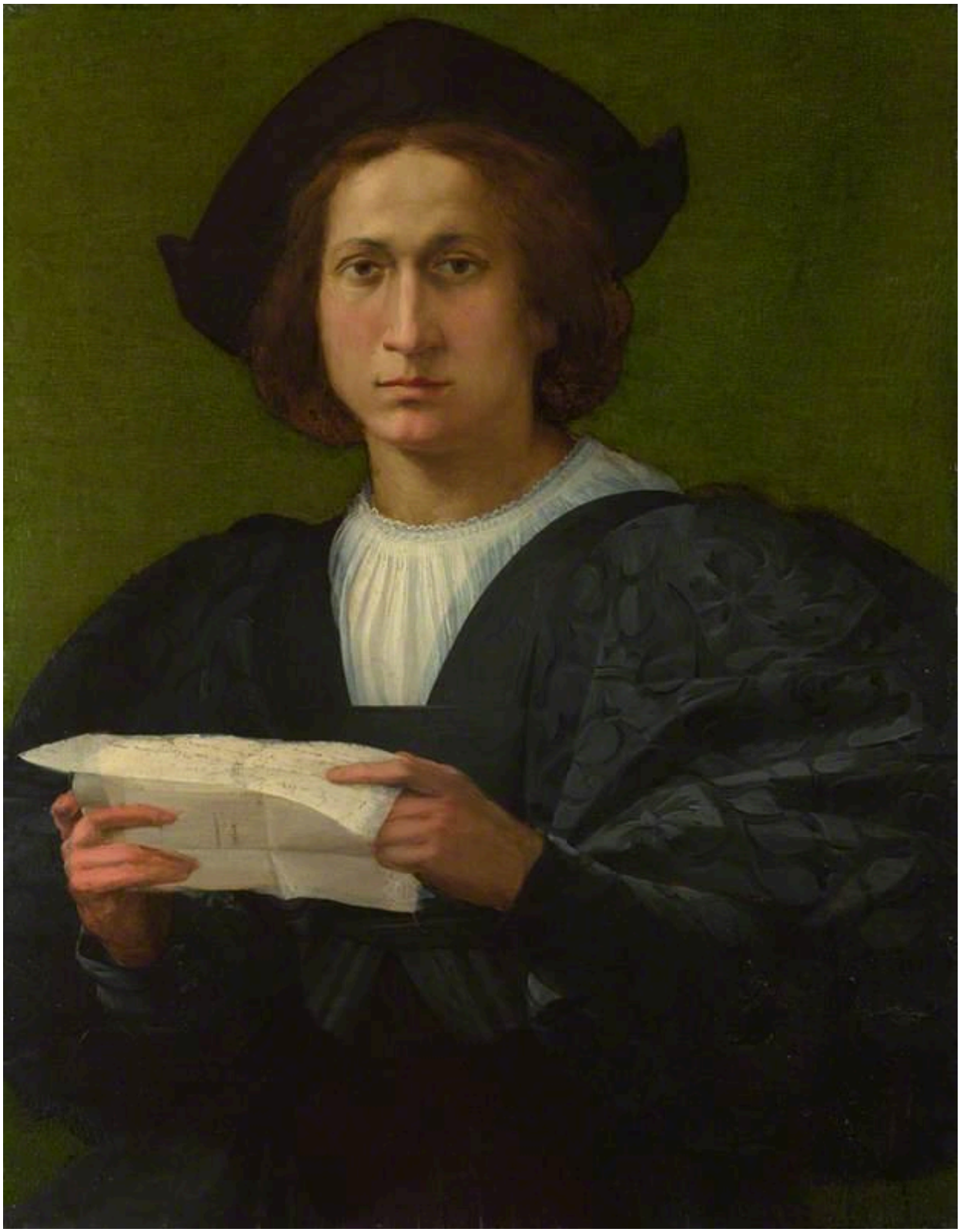
1. Know your users. User research and testing with the right person is key!

Knowing the target users and listening to them is the most important part of the design process that we do, but reaching out to the users with the right

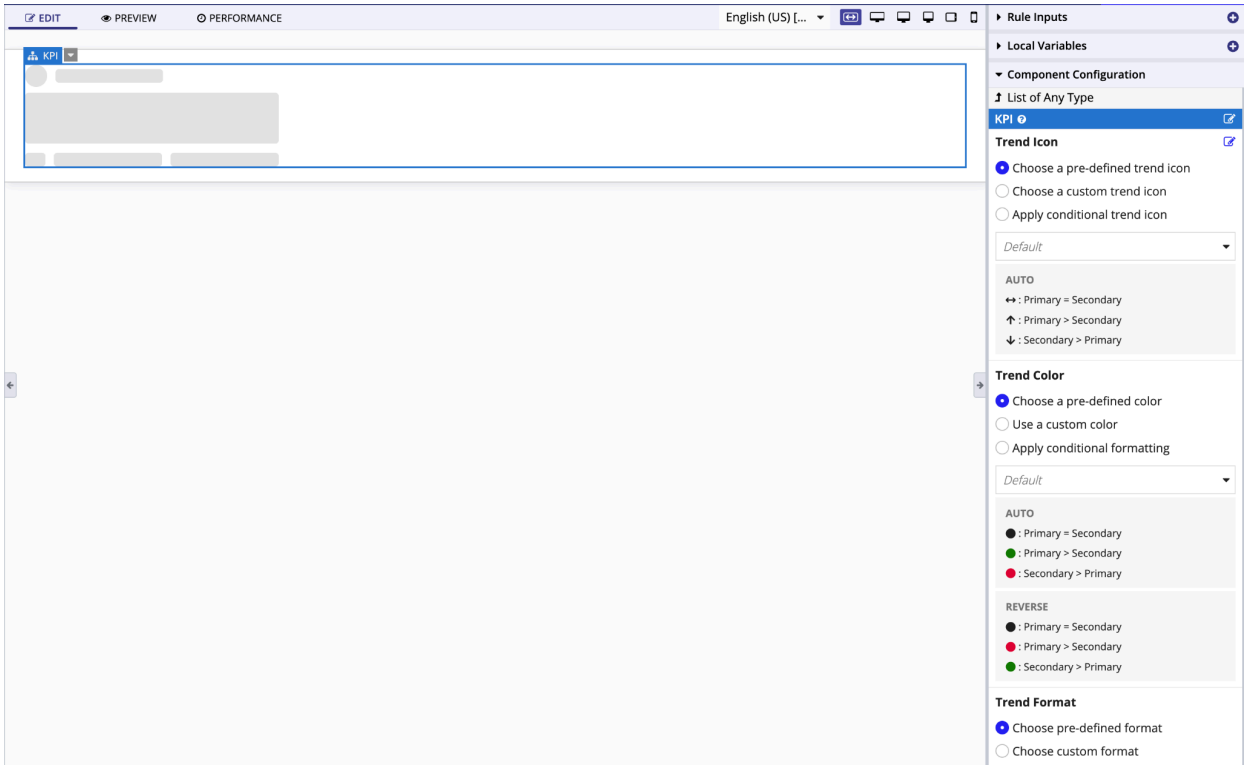
knowledge and experience is the key while designing a low-code/no-code developer tool. For example, addressing a simple documentation copy change based on a user feedback during the usability testing helped us to enhance the overall experience of the component configuration. So, A person with right knowledge and context is very important because they have the business context of the product and help us to make the experience better for all those who will use the product. Unlike other consumer applications that allow users to perform specific functions, developer tools cater to users who have complex goals. Low-code and no-code developer tools are even more complex because they're often used by individuals with limited knowledge of traditional software development languages, who rely heavily on visual builders to create products.

When designing these tools, conducting multiple user research studies and user testing is crucial. Accessing your users (developers, consumers, etc.) should be the first step before beginning the design process. User research provides insights about aspects developers may not be aware of, such as what low-code developers notice with end customers. The primary goal of all developer tools is to improve developers' efficiency in building and delivering products quickly to customers. Through user research and testing, you can understand developers' working patterns and focus on enhancing their overall experience.

2. Provide users with the ability to access data and documentation.

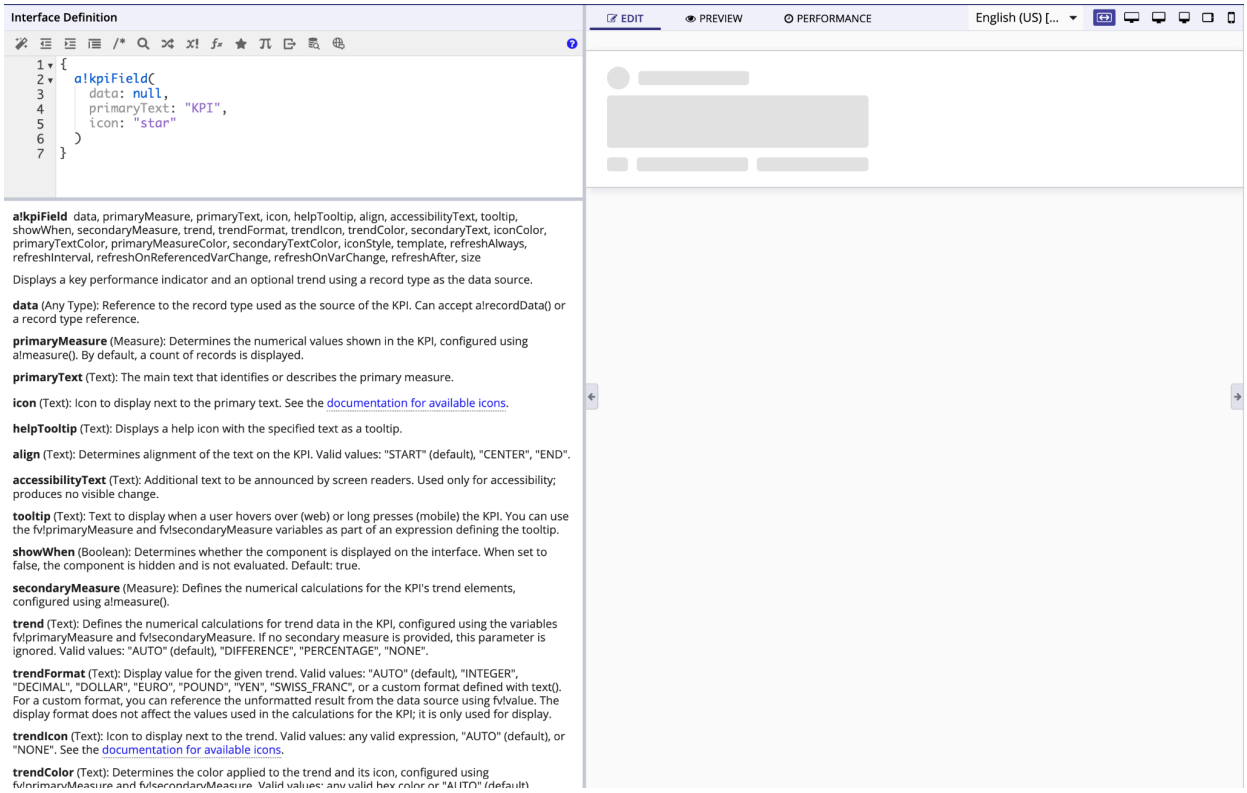


Access to data and documentation is crucial for low-code developers utilizing the platform. When developers build a product, they must switch contexts between various tasks, requiring access to data, object search, creation, and more. Constantly switching between screens can decrease productivity. Thus, providing the ability to access data and create objects directly from the builder would enable developers to focus on their tasks.



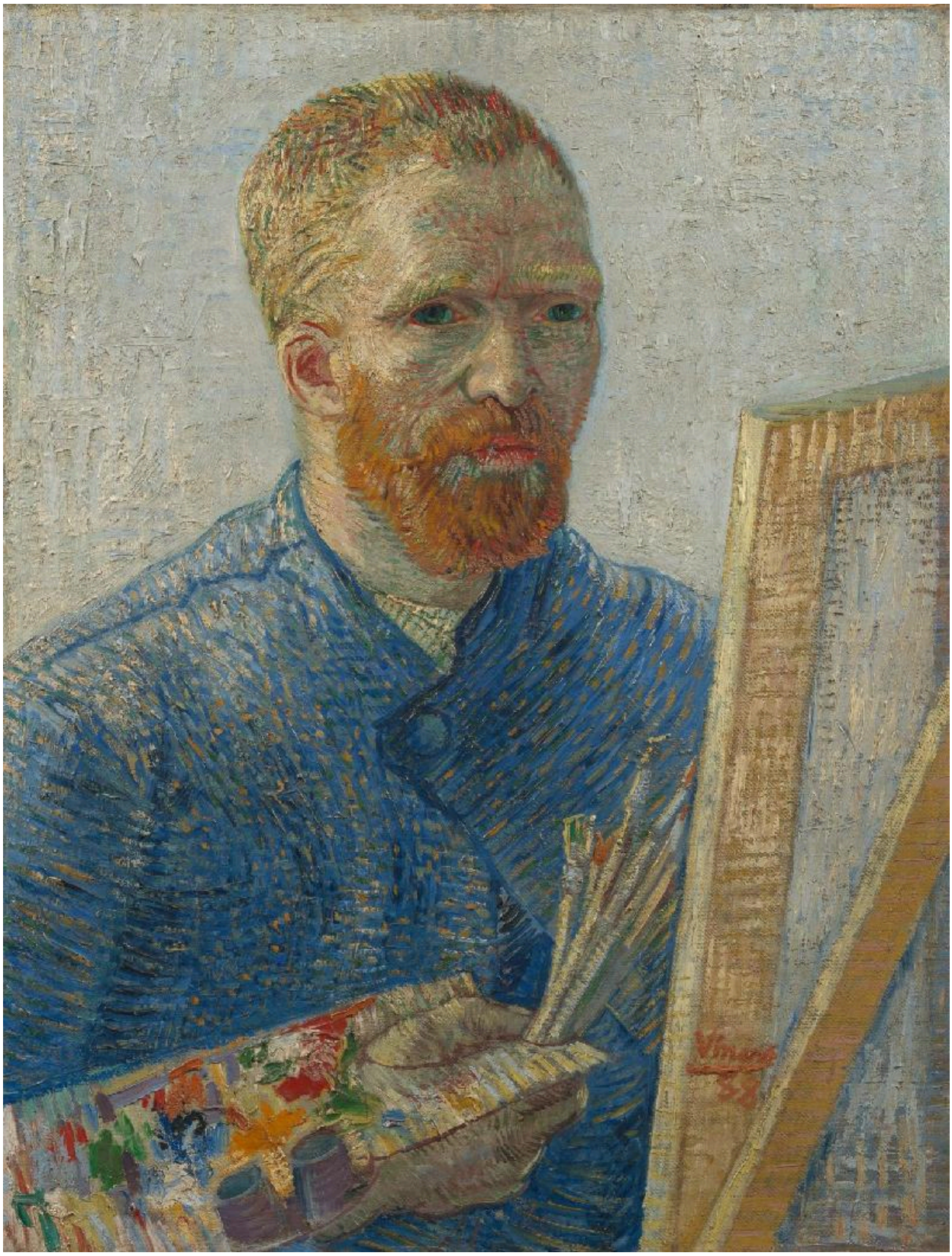
Legend for users to understand the predefined values

Documentation is equally vital for users. Low-code platforms are often utilized by individuals who lack a complete understanding of the platform and its components. Offering a clear, guided experience for building an application is essential. Users should be able to access the platform's documentation at any time to gain insights into components and their configurations.



Component documentation in the interface designer

3. Design tools don't matter!



As a product designer, we often become accustomed to using a particular design tool rather than focusing on the overall design process. Like many designers, I too believed that design tools such as Figma or Sketch were the most critical aspect of my design process, and that they enhanced my design skills. However, upon joining Appian, I quickly realized the importance of unlearning certain habits and familiarizing myself with their SAIL Design System, which users utilize to build applications. I had my own concerns, such as whether transitioning away from Figma would diminish my relevance in the design industry but on the contrary, it allowed me to learn more about the design to development process and provided me with a better understanding of the overall product before jumping into design. Nonetheless, this experience helped me understand that the process is what truly matters, not the tool itself.

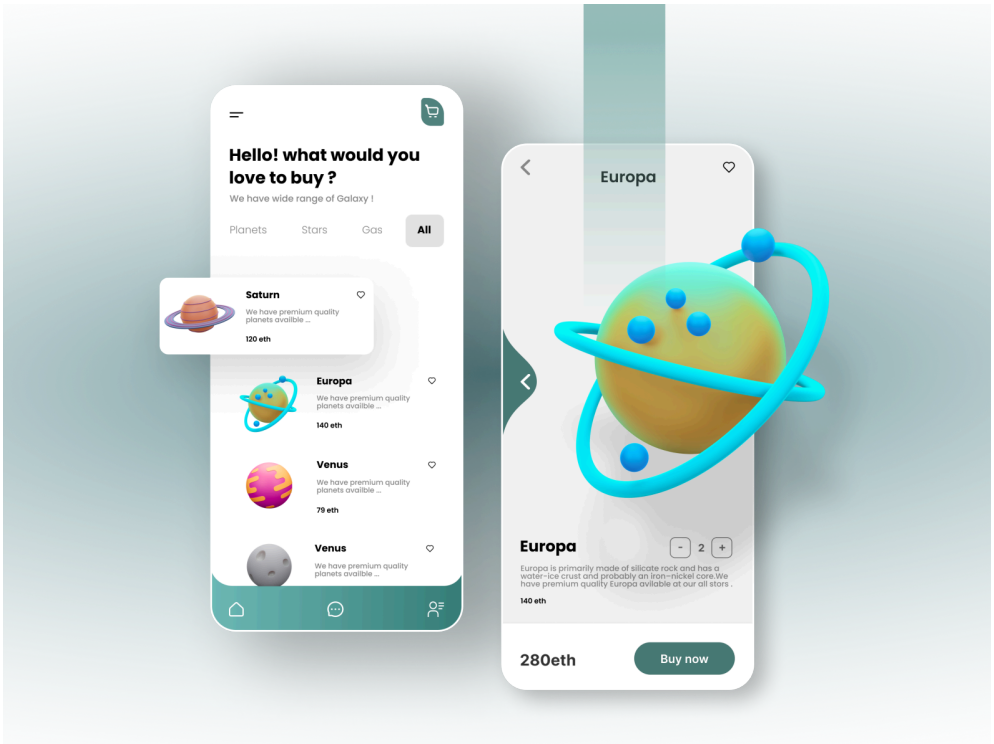
Every year, numerous design tools are introduced, and individuals often switch to new ones based on their requirements. Thus, as a designer, it's essential to prioritize the process and creativity over the tool. Whether designing in Figma, using pen and paper, or a low-code design platform, the medium is inconsequential. In the context of Appian, where we cater to low-code developers, it's crucial for designers to comprehend the SAIL design system and adeptly utilize it in their work.

The key takeaway for me has been to

- Cease reliance on tools for creativity.
- Select a tool based on the specific use case or personal preference.
- Continuously expand and refine one's design knowledge.

4. Just because it looks beautiful, doesn't mean it's the best solution

Fancy user interfaces often take precedence in design discussions. While many products successfully marry user experience and visual aesthetics, some prioritize visuals over overall user experience. As designers, we sometimes prioritize building attractive user interfaces without adequately addressing the overall user experience.



Unrealistic dribbble example

Developer tools can and should have a great user interface, But prioritizing a seamless user experience over flashy interfaces is crucial. Developer tools typically carry a heavy load of information, requiring minimal distractions on the screen for easy access. Our focus should be on enhancing accessibility and readability by reducing visual clutter and unnecessary dialogues. Developers should easily access necessary features to swiftly build applications, which aligns with the primary goal of low-code or no-code platforms.

Users utilizing these platforms for enterprise product development may not prioritize visual aesthetics but instead concentrate on product functionality. Providing them with out-of-the-box visual patterns for different use cases

can facilitate their work and improve their product's visual appeal without relying on external designers.

The process is rewarding!

Designing for a developer platform presents a real challenge. It's a complex tool with a wealth of information. Consequently, the process might seem less creative, but these constraints compel the designer to arrive at an ideal user experience for a developer tool. Developer tools are crafted for building products. The most rewarding aspect of this process is witnessing low-code/no-code developers leverage the tool to create outstanding products for various use cases.

Thank you once again for reading, and I eagerly anticipate your feedback and comments.