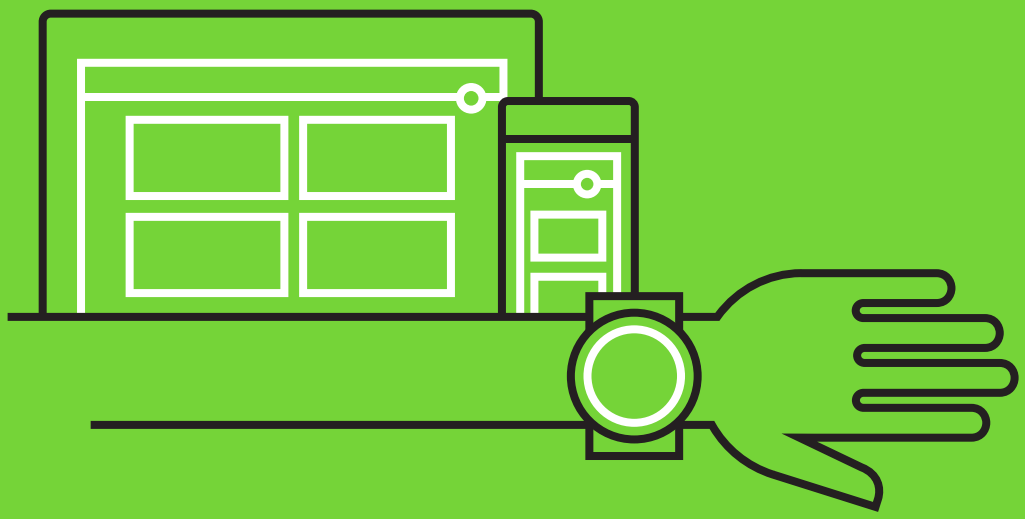


MOBILE APP DESIGN THAT SCALES

Building for Existing and Future Devices

Telerik®

CONTENTS

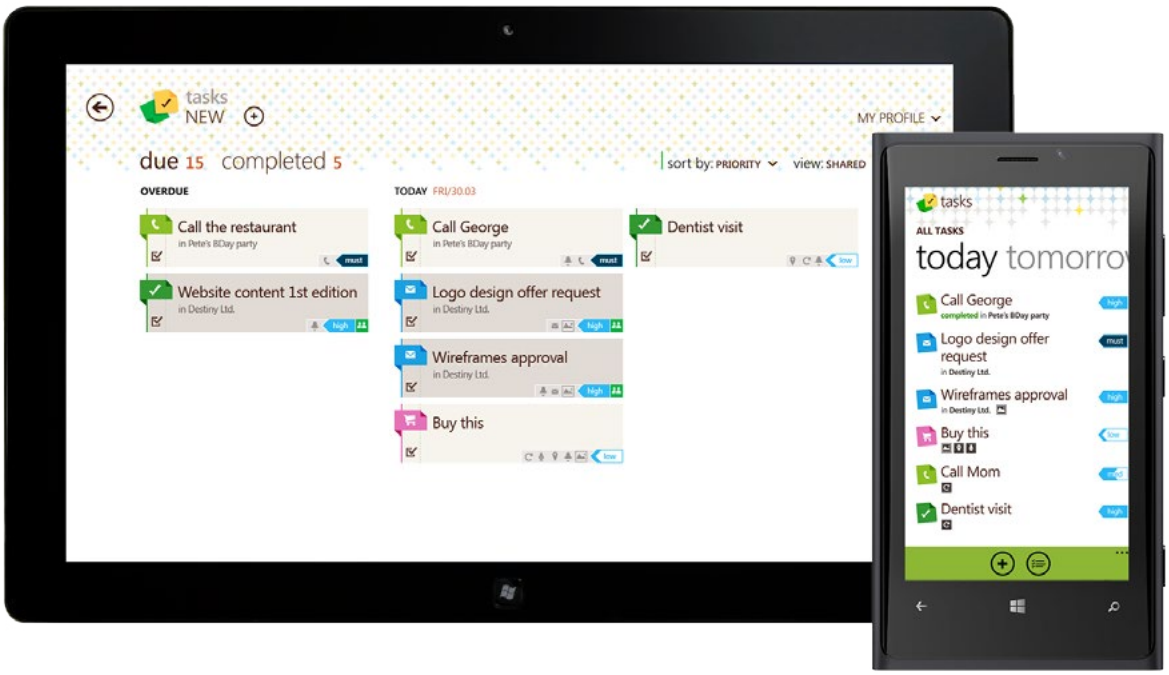
INTRODUCTION

Mobile designers and developers share a common frustration stemming from the fragmented device landscape. With wearables in the game, the puzzle is becoming more baffling. Is it possible to design the perfect app and user experience for all devices, from TV to wearables? This whitepaper offers practical step-by-step guidelines to help you design flawless cross-channel app experiences.

THE APP

Since the announcement of Android Wear, we had plans to redesign our Tasks application for the Android ecosystem. The announcement of the Material Design System made it obvious that we'd have to reimagine Tasks from scratch. Tasks is a "to-do" lists app originally designed for Windows Phone and Windows 8.

For our study, we chose a more simple scenario than the one in the actual app. This presumes the principles we'll validate would then be applied to more complicated scenarios. We stripped down the task to its essentials: name, due date, reminder, priority and picture. The tasks could be organized in categories. We were reimaging the app, so we planned as though the Windows Phone/Windows version didn't exist.



ADAPTIVE LAYOUT IS A NO BRAINER RIGHT?

When we think about design, responsive design and adaptive layouts are a must given the diversity of form factors. Adapting the layout to different screen sizes starts with identifying breakpoints. A breakpoint is a point at which certain aspects of website or an application change depending on specific conditions. The breakpoints we chose for Tasks used the Android classification and were measured in device independent pixels–dp:

- Small and normal devices: Less than 480 dp. This accounts for most phones
- Large devices: Between 480 and 720 dp (i.e. Nexus 7")
- xLarge: 720 dp and above (i.e. Nexus 10")

> Adapting the layout to different screen sizes starts with identifying breakpoints.

WHEN ADAPTIVE LAYOUT IS NOT ENOUGH

These breakpoints cover the current device landscape in Android. We would create a different layout for each of these screen sizes, both for portrait and landscape orientations, if necessary. But what if a radical new form factor or platform is introduced, like wearables? To redesign and develop again from scratch is not a practical business solution. What happens when adaptive layout is not flexible enough to make the design fully responsive? Content needs to be structured to work on every possible screen size—existing and future—so we should intentionally design that way. Everyone is in the business of creating content, even Tasks. Users not only want access to their tasks, but also to have the ability to create them on any device of their choosing. Future platform-proof core content will be the foundation of our app. We have to be prepared to add the presentation layer on top, depending on the context, device and use case, without changing the core structure in the process. **Our app will be platform-agnostic for content, and platform-aware for the user experience.** There is no primary platform—all platforms are equal.

To make our app design truly responsive, we added to the need of adaptive layout, the need of adaptive content. In Karen McGrane's talk, "Adapting ourselves to adaptive content," she gives wonderful examples of how to develop a content strategy. While she focuses on large content publishers, adaptive content is totally applicable in situation like ours. It's the right approach if imagining your app will run on all platforms, whether mobile or tablet apps, mobile or desktop web, social media or anywhere else—even print.

Adaptive content is:

- **Structured:** Consists of bite-sized pieces of content
- **Designed for re-use:** Has multiple versions for different contexts
- **Designed as a service:** With APIs, data storage and future concepts for how people will interact with it.

Let's get back to our concept app. We've already determined our breakpoints and structured our content according to the principles listed above.

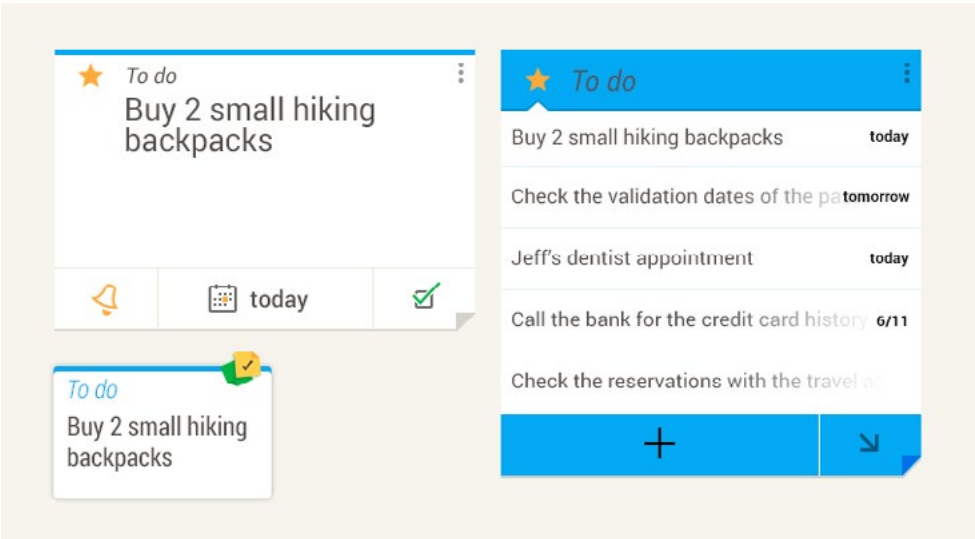
Now, we face our next design challenge: users want to share tasks in progress with others– for example on their Facebook family group page. We need a user interface solution that not only combines the adaptability of the layout and content in the most beneficial way, but also creates rich interactions in a consistent design format that works across multiple platforms, including web, mobile, apps, wearables and more.

CARDS

You may already know what cards is if you've stumbled upon apps like Tinder, Google Now, Twitter, Jelly and so on. We've already written on that topic.

According to the Material Design system guidelines, a card contains a unique data set of related, heterogeneous information, for example, a photo, text, and a link all related to a single subject. A card delivers information in an easily readable way, and card interaction comes naturally with a single finger (usually the thumb on the phone). That's why it is perfect for smaller screen sizes like phones and wearables. Cards combine an information design pattern with their own interaction model, a set of gestures, which creates an engaging user experience.

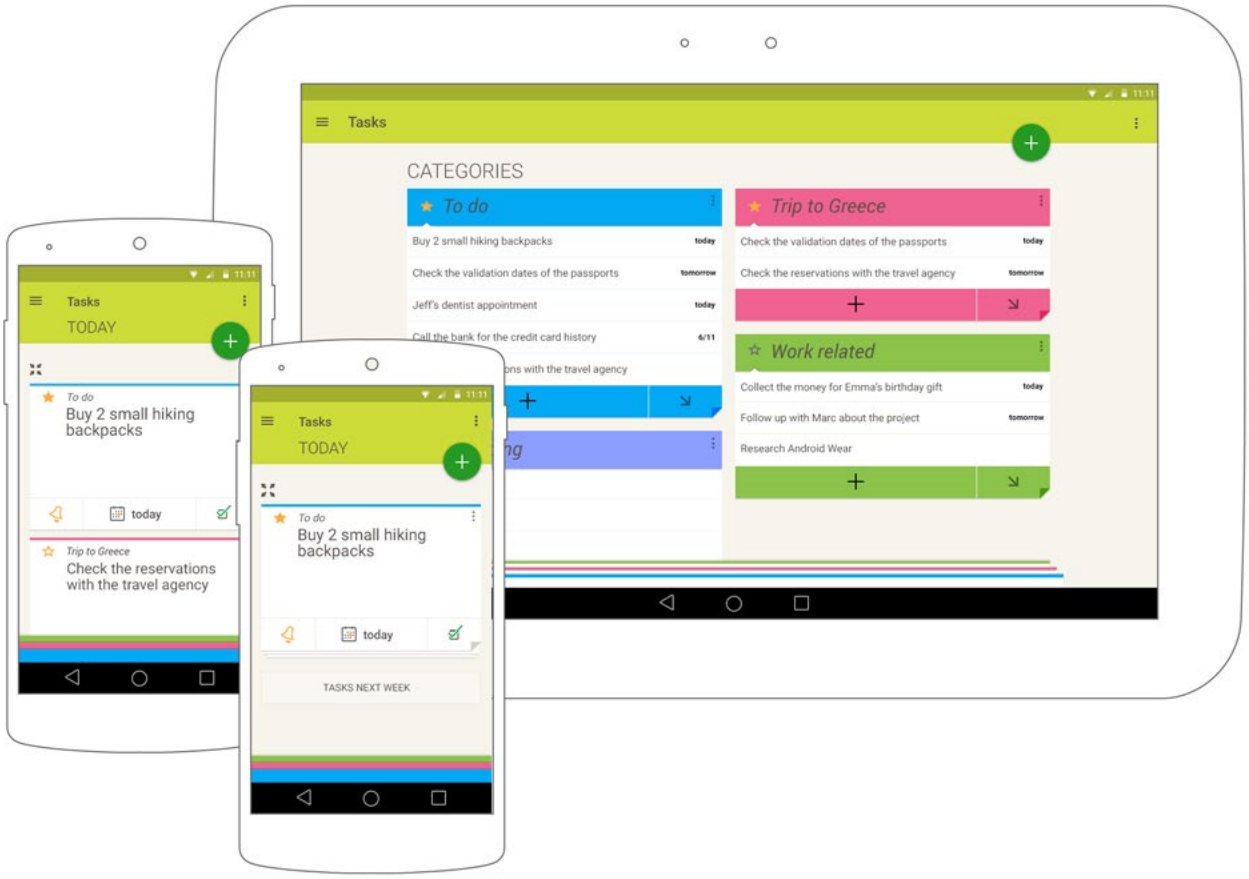
> Cards combine an information design pattern with their own interaction model, a set of gestures, which creates an engaging user experience.



We chose cards as an atomic unit of our content for several reasons:

- **Contextual structured content:** A card can contain a single information item together with a set of attributes or actions related to the item, which provides the user with choices in the context of this piece of information. This is how cards distribute context.
- **Scalable:** Adaptive by nature, the layout configurations are very flexible. They can be stacked like deck of cards, or organized in a list or grid view.
- **Fun:** A single card can be swiped, marking the task as done.
- **Sharable:** Design and information consistency remains intact on a variety of platforms.
- **Meaningful contextual information:** Cards can show meaningful information depending on the context. The information displayed can vary both in terms of length and content – the answer to the user's question is provided before the question is asked.

At this point, we need to pay attention to the UX principles of the platform for which we are designing.



MATERIAL DESIGN SYSTEM

The Material Design system can be explained with three words: light, surface and motion.

Light: Google gives developers and designers a built-in way to provide depth through the z-axis and shadows of the user interface elements.

Surface: The material metaphor is a way to see every piece of UI as something tangible and familiar. I won't get into details with the paper metaphor which is extensively covered in these brilliantly written guidelines.

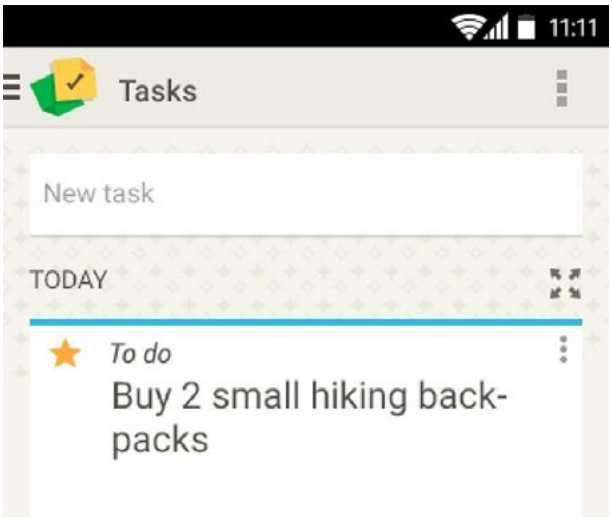
Motion: The usage of animations and meaningful transitions provides context and affordance.

With these principles Google, unifies the user experience across the variety of devices and form factors. **As the device count grows, the biggest promise of the Material Design system is the transition of content between them to look and feel natural and familiar.**

How the Material Design guidelines are applied in our app:

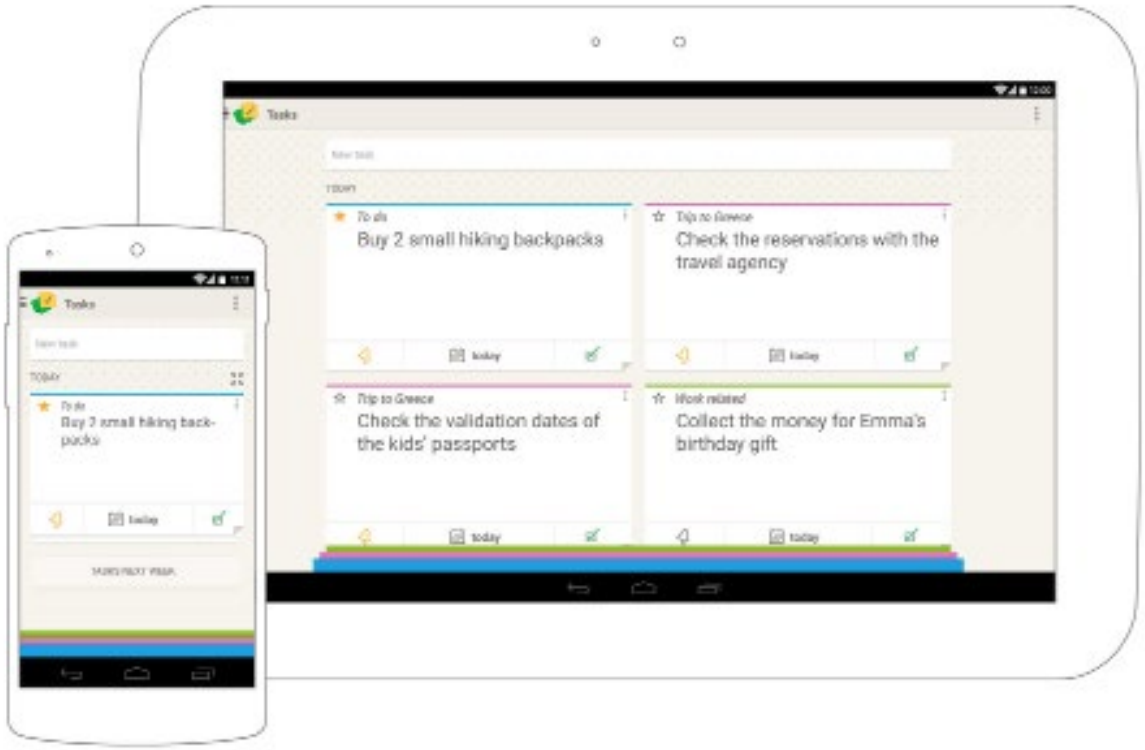
We started designing the app before Google I/O 2014. After Material guidelines were announced, we had to adapt the design once again. The core design principles and Tasks UI elements remain the same: the adaptive layout, the adaptive content and the cards and they are supported by the Material Design system by default.

1. Typography, space and color.

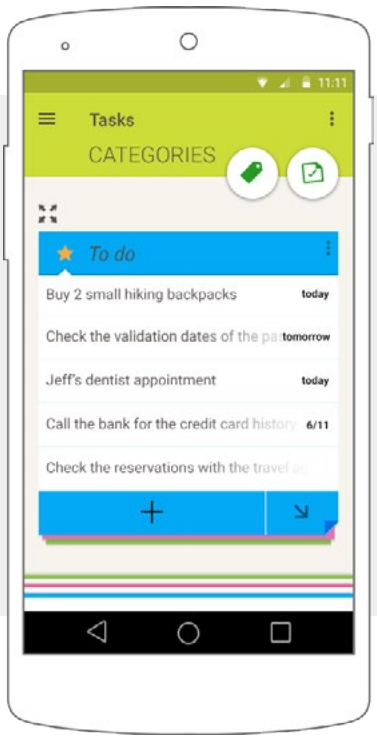


Square baseline grid—8dp for all elements, 4dp for type and iconography—margins according to the guidelines. Bold colors and typography make the initial design background unnecessary.

2. Floating action button (FAB)

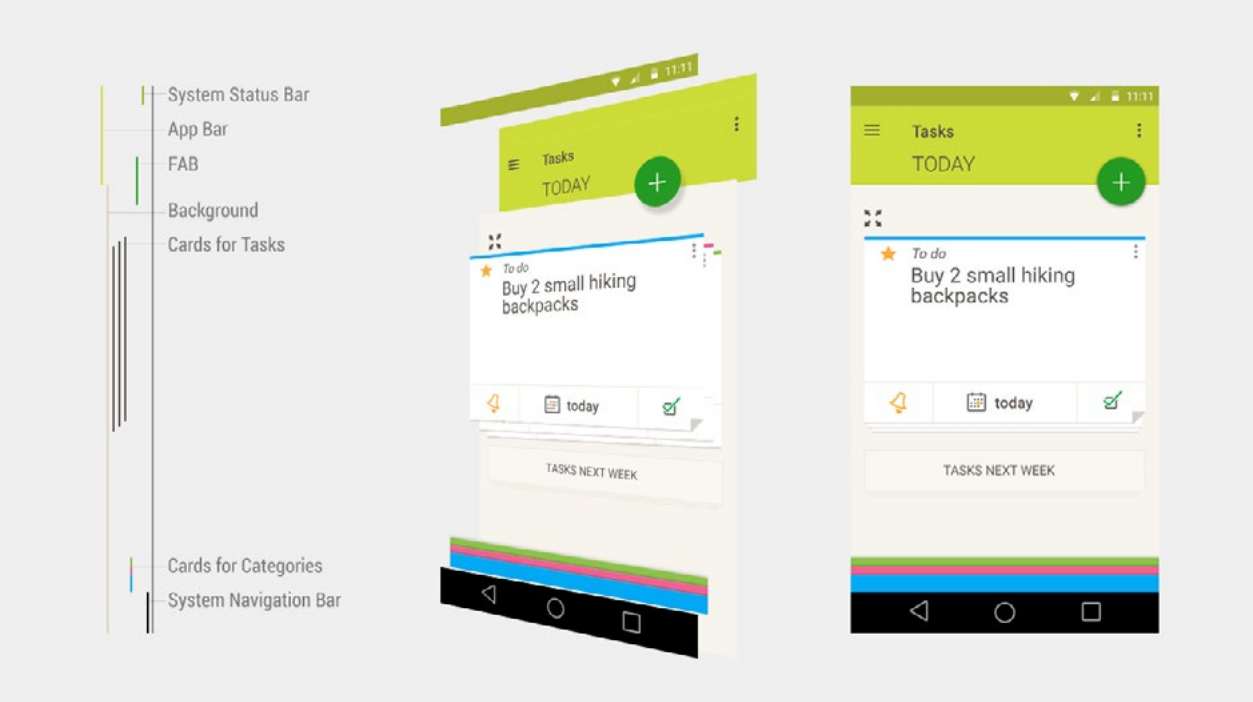


In the categories view, the FAB morphs into two related action buttons—new task button and new category button—since both primary for this context.



3. Depth

Although depth is not a visual ornament and it should support the information architecture, we added it here to emphasize the meaning.



4. Cards

Our cards vary from the Material guidelines a little bit, but here is what we thought: we stepped on the solid base of these guidelines and created a solution that solves the user's problems in the most effective way without compromising the Material Design principles. It is an improvement.

5. Animation and delightful details

Most of the points above are only the designer's concern, but there is serious work to be done by both designers and developers before everything is polished and ready to ship. That's why Material Design is something more than a set of principles: it is a team collaboration.

At this point, we can easily state that the design of Tasks for Android is almost done. Following the Material Design system, we are being consistent in terms of transitioning the visual presentation of the content for Android Wear. **What's left to decide is which part of this content makes sense on wearables.**

> Material Design is something more than a set of principles: it is a team collaboration.

WEARABLES

There is a big conversation about whether people use it or abandon it over time. That said, if we understand the experiences users want out of these devices, we can tailor content that makes our app relevant for wearables.

The superpowers of wearables are:

- **Sensing:** The information they collect should be meaningful to their owner.
- **Personal assistance:** Wearables should provide a set of context-aware and timely notifications, without overwhelming the user with frequent interruptions.
- **Fast Access:** Information is on the wrist, in front of eyes, and accessible without reaching for a phone or other device. You can take a picture instantly with a wink of an eye, making wearables ever-present at just the right moment.

The most important thing our app does is to help users organize their time. To do that, it should remind the users for their tasks in a contextual way. Getting contextual triggering right has great impact on the user experience. Each of the wearable superpowers above is about context: the right information for the user, the right time to display a notification, resulting in the most reasonable action for the moment.

> Getting contextual triggering right has great impact on the user experience.

We have two types of notifications:

- **Just in time notification:** for tasks with fast approaching due dates
- **Just in place notification:** for tasks which can be done near to a users' location

These notifications will appear in the context screen of the Android Wear:

On card dismissal, the task is considered done. Otherwise it can be postponed with the action button. For just in time notifications, it automatically postpones the task one day. For the just in place notifications, the task is postponed until the next time a user passes the triggering location. The combination of both is not possible by design.

Just as important as the notifications, is the tasks creation. The cue card in Android Wear enables a user to speak to Google. Our app will register to a voice intent users will use to create tasks for a specific due date, or for the location they are at the moment.

It is as simple as that. Just carefully consider what would be relevant for the user's context (time, location, physical activity) without being pushy. These two simple rules will make the app a helpful addition to users' everyday life anywhere—on their wrists or in their Google glasses. As stated in Designing for Wearables in Google I/O 2014, the notifications in Google Glass and Android Wear will be shared.

CONCLUSION

It looks like we are ready to answer the question we asked in the beginning. Is it possible to design the perfect app and the perfect cross-channel experience on a mobile phone, TV, on a Twitter or Facebook summary card, wrist watch or even a refrigerator?
With our approach:
the adaptive layout and the adaptive content strategy backed up with the cards—we're close enough. By looking for the answer, we reinforced that an app needs to be relevant above all else. That means it should have a relevant interface, show relevant information and deliver a relevant user experience. Today, you can't have a relevant app if you design only for just one use case, device and context. Regardless of the approach and design process you choose, creating design that scales across all existing and future form factors and platforms makes an app relevant to the most users.

Creating design that scales across all existing and future form factors and platforms makes an app relevant to the most users.

ABOUT THE AUTHOR

Nina Zayakova
UI & Interaction designer

Nina Zayakova is a UI & Interaction designer for mobile in Telerik. Her growing expertise in designing for different mobile platforms is driven by constant curiosity and urge to help people understand. She strongly believes that user experience should never be compromised. You can read more from Nina on Twitter at @myninka.

Telerik helps developers create engaging mobile experiences regardless of the coding language of choice. Check out some of Telerik products helping you build mobile apps faster.

Native Development

UI for Android
for building pure native Android apps using Java

UI for iOS
for building pure native iOS apps using Objective C

Cross-Platform Development

Telerik Platform
for developing cross-platform and mobile applications

NativeScript
for building cross-platform native apps with JavaScript

UI for Xamarin
for building iOS, Android and Windows Phone native apps with C#

Hybrid Development

AppBuilder
for building cross-platform hybrid apps using a single pure HTML5, CSS and JavaScript codebase

Web Development

KendoUI
for building modern sites and apps with HTML5/JS

Telerik®
Develop experiences