

Top reasons apps fail design review

Let's start at the top – panorama design

They're like a magazine cover experience; they highlight content and give an overview of the application

Don't put all your content or functionality on them – they will look cluttered and performance will take a hit

Pick three themes and showcase them

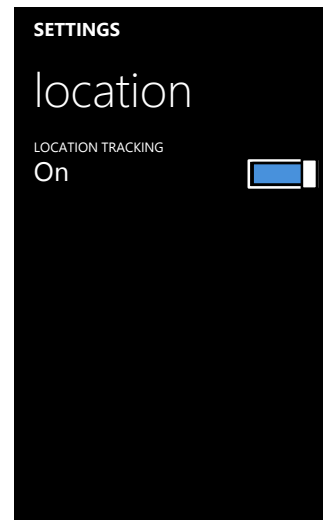
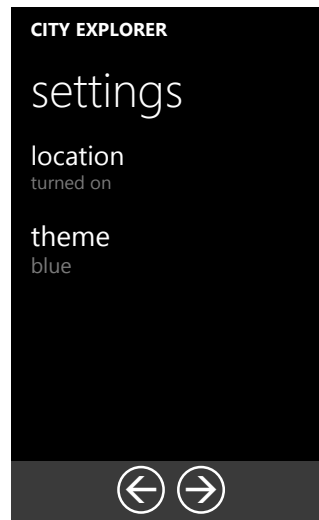
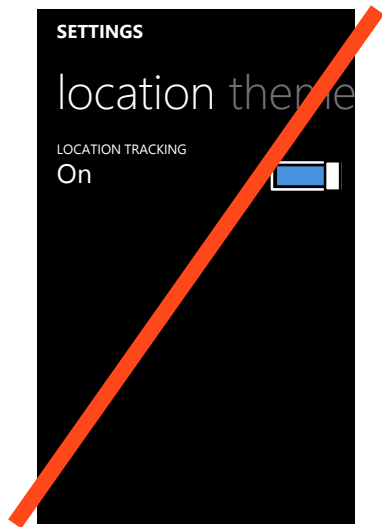
Use a maximum of five panes

Even better:

Make them immersive; use background imagery and brand colours to bring them alive

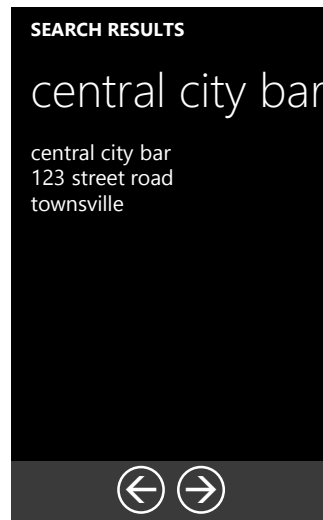


Say no to competing for gestures



Never put controls onto pivots (or panoramas) that require a sideways swipe gesture, as these compete with the swipe gesture to change panes.

Say no to massive pivots



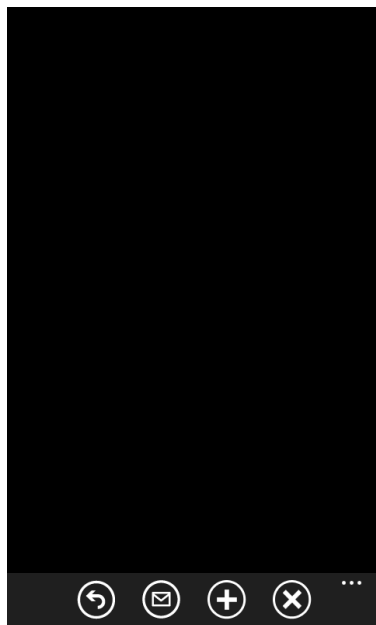
Aim for a maximum of 5 pivot panes in a pivot control unless the titles are sequential (e.g. days of the week, months of the year). Even so, keep the number to a minimum for performance reasons.

Then: buttons, buttons, everywhere

Buttons that perform actions should always go in the application bar. Don't let them float around the UI; it looks cluttered and messy.

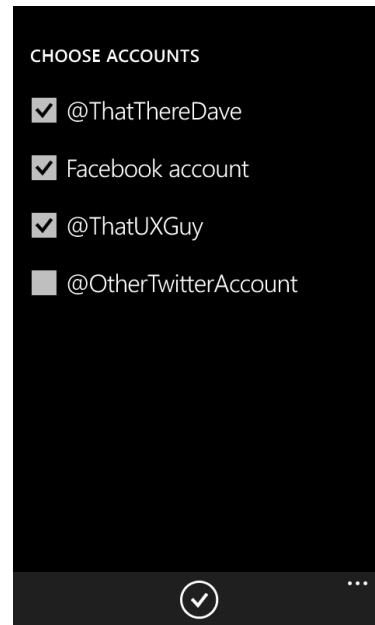
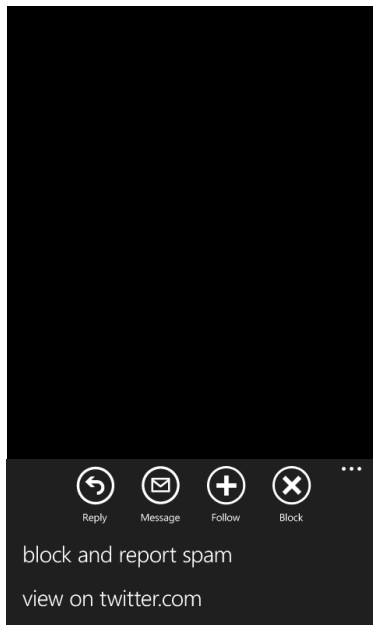


Application Bars



Primary actions

Secondary actions

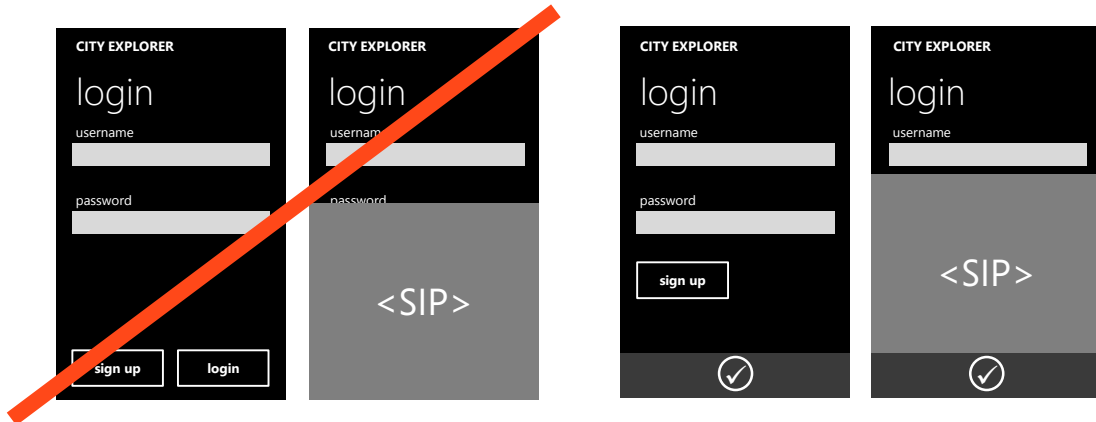


More buttons, buttons, everywhere

On log-in pages, the SIP often covers up the login button.

Placing it in the application bar will speed up the process for the user by ensuring it is available when the SIP is onscreen.

Use the right SIP to support the login action.



A common exchange

'My users won't notice the buttons in the app bar'

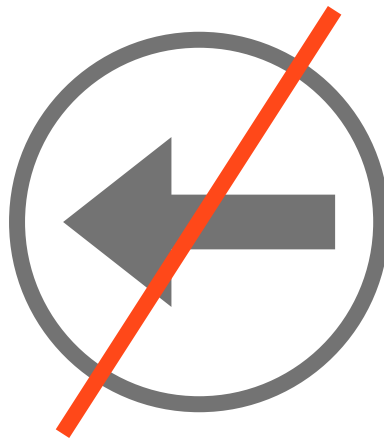
Not true.

Windows Phone users are used to going to the app bar for actions, so they are just as discoverable here

Do not use these buttons



Close buttons



Back buttons

This one is tricky

Very few apps need a home button. Users rely on the hardware back button for navigating up the back stack.

Creating buttons that navigate around the UI will interfere with the back stack and confuse users.

Only apps with unique architecture justify the use, such as shopping carts (at the end of the transaction), or complex task orientated apps.

Otherwise, needing a home button is usually indicative of an issue with the information architecture of the app.



Web browser controls

Please don't use web browser controls.

They're demanding on memory, slow to respond, and the gestures often compete with those of other controls on the page.

Fixing these issues can take a lot of time, so focus the effort on creating a web service and put Silverlight controls on the screen instead. The experience will be much better and your users will thank you.

Failing this, take the user to Internet Explorer. Make it clear to them that this will happen.

If you must use a browser control, disable panning and zooming.

Porting an iOS/Android app

The application bar should not be used like the tab bar (global navigation) on the iOS and Android platforms. These are different paradigms.

Equally, some controls don't exist on iOS/Android that exist on Windows Phone, and vice-versa, so trying to port metaphors from other platforms could result in a poor experience.

Ensure that you align visually with Windows Phone too. Don't try and make your app look like the iOS or Android version.

A common exchange

'We want consistency between our mobile apps – the iPhone, Android, and Windows Phone 7 apps should all look the same'

This won't work out well for the end user.

Windows Phone 7 is a markedly different platform, not only in look and feel, but also in interaction metaphors. Contorting an iPhone app to fit into Metro will result in an app that is confusing to use, difficult to navigate, and visually inconsistent for Windows Phone 7 users. We can work with you to help you align and land your brand appropriately.

Designing lists

Don't use arrows to indicate a list item is clickable.

List item



List items should not have a visual selected state.

Remove borders and backgrounds from list items. Use text size, colour, weight, alignment and spacing to visually group information instead.

Branding

We want the user's brand to shine through, however, we don't want it to overpower the user. When in doubt, talk with us about it. Some tips:

Good ways to brand the experience

Use a background on the panorama to make the screen more immersive

Add a brand logo to the panorama, or a smaller one to the top of pivot controls

Use the brand colours as accent colours in the app

Bad ways to brand the experience

Use a custom typeface throughout the app

Use custom styled buttons instead of standard Metro ones

Use background colours on titles and list items

A common exchange

'Our brand is very important and we need it to shine through'

It can.

But emphasise your brand in the *right* places with text foreground colour, panorama backgrounds, small brand elements at the tops of pages. Don't use your custom font – this is making the user's experience of their own phone inconsistent.

Tombstoning

Take care to store and reload the contents of edit fields (and other controls) on a page when tombstoning.

Return to the same pivot page or panorama position as the user left off.

Give the user feedback

Make sure you give the user some visual feedback when they touch something.

The standard feedback is the tilt animation, but do not apply it to controls like checkboxes, toggle switches, etc.

When loading content show the indeterminate progress bar (the chasing dots) if you don't know how long it's going to take.

Make it snappy

Try not to reload content from the network every time a page is displayed – it irritates users. Caching is everyone's friend.

It's preferable to load a stale state quickly and run a background refresh with a seamless transition than show nothing until the refresh has happened.