Solarizr10 Design

Initial style considerations

Color

Solarizr 8 (and many of the original modern UI styles) were quite dark. One of the first choices I made was to lighten the color palette to reflect the simplicity of the MDL2 design language. The minimalism in MDL2 often involves juxtaposing borderless elements, so having a few different background colors helps when defining panes and other juxtaposed content.

I settled on the blue and white palette to reflect blue sky and white clouds. Blue and white are calm choices for large areas and are a good foil for the brightness of the logo, which is yellow and orange for the sun. The hamburger menu is a brighter blue than the background blue because it is an action item.

Font and Icons

The MDL2 Assets icons are beautiful, simple, and coordinate with Segoe UI. I feel like they are always a good choice when you want to subtly brand your app as a Windows app, and yet they are universal enough to work with any app design.

Branding

Solarizr 8 (and many modern UI apps developed for Windows 8) used a background image to brand the app while preserving the simple modern UI style in the content foreground. Background images run the risk of being too literal and not adapting well to different screens. For Solarizr 10, I wanted to preserve the solar panel grid as a branding item. I created a vector grid using the iconic solar panel shapes as a more abstract version of the Solarizr 8 background image. That grid now shows up on the login page — it would have been too cluttered to use it anywhere else with the new layout we have.

Layout and inspiration

For inspiration, I looked to the News, Outlook, and Skype apps.

News

I liked the use of the navigation in the News app. It uses what is more or less the “canon" version of the SplitView navigation paradigm for Windows 10. When designing, we walk a line between following convention and creating something unique. Navigation is one of the most important conventions to follow on a platform to make it easy to interact with your app.

A major navigation choice for Solarizr10 was pivots vs. SplitView items (or a combination of both). Our main Solarizr views (Appointments list, Map view) were previously hub sections in Windows 8. They are basically different ways to get to the same information, so it makes sense that that they be more “global” navigation items that you move between less often. The user chooses which view he wants to interact with on the SplitView, and then pivots can be used within each view for related content if needed.

Skype and Outlook

Skype and Outlook both use Master/Detail panes. In Skype's case it's pretty simple — the left pane has contacts and the right pane has conversations or calls with that contact when selected.

Outlook is really interesting for UWP because it has so many optional panes: folders and tool navigation (Mail, Calendar, etc) are on a custom SplitView that collapses for smaller screens. The list of emails is the main pane that needs to remain available no matter the screen size. The detail pane for the selected email shows up on the right if there is room. That for me inspired the use of real estate in Solarizr. We show what is essential: the list or map of appointments, and bring in a detail view if we can.

This is probably a good place to mention state, which I should have thought about at this point in the design. The problem of which appointment to show in the detail pane at any given time would show up if we did full-blown flows for a real-world app.

The Master view and Dashboard view

The only way to access appointments in Solarizr 8 was through the initial hub section that flowed appointments around a hero in a grid. But that section was not endless, so we were limited to only showing the most recent 20 or so appointments. Solarizr 10 solves that problem by using views that have the ability to show all appointments.

The Dashboard view is a nod to the Solarizr 8 Next Appointment hero. It is still useful to have an initial view that gives you a quick glance of your upcoming appointments and highlights the next one for quick and easy access.

The Detail view

Solarizr 8 used a three step process on Mobile to edit the appointment. We showed one section at a time – Appointment Info, Add photos, and Comments/Status were all separate steps. On tablet/desktop, we used hub sections for those steps.

The content entry is so simple for Solarizr that it made sense to combine it into one step, which works well with the master/detail view and UWP. We can now provide a more consistent experience across devices by encapsulating the appointment control. Whether you see it in a pane on desktop or as the full view on mobile, it is the same layout.

Panes on desktop are great because they can port so well to a portrait view on mobile.

Iterations

Starting to build quickly was an important part of this design. I put in the "knowns" like SplitView and master/detail panes with VisualStates for different devices. Having a clickable and resizable prototype made it easier to fill in the panes with realistic layouts.

I quickly realized that some design elements (like the carat that made the detail pane look like a speech bubble) were not realistic. The calendar view was also not realistic to build in the scope of the project, so it made more sense to use calendar integration.

My original designs extended the view into the titlebar. The simpler branding of Solarizr 10 meant that you don't see the logo or app name on every page, so in this case leaving the titlebar intact helps to identify the app.