Solarizr Specification

Overview

This document describes the scenarios and context for the Solarizr Universal Windows Application. The Solarizr app is intended to be used as a reference application to facilitate guidance on how to build non-trivial applications for Windows 10. The application will not be completely "store ready;" however it will use best-practices to demonstrate key strategies for building similar applications.

Solarizr – The Company

The Solarizr company is imagined as a vendor for solar panels that provides services to survey potential installation sites and to then install solar panels. Imagine the scenario where a homeowner decides they would be interested in determining if their home is a good candidate for the installation of solar panels. Such a homeowner would call Solarizr and schedule an on-site survey of their home. The Solarizr call center would schedule an appointment, assign an engineer to perform the survey, capture the survey data and inform the homeowner of the outcome.

The Solarizr Universal Windows Application represents the engineer's view of this process.

Key Scenarios

The Solarizr application is broken into two parts utilized by two key personas – a call center persona and an engineer. The key scenarios imagined for each of these personas is outlined below:

As a call center operator I:

- Wish to be able to schedule an appointment for an on-site survey. I need to be able to collect:
  - The customer name.
  - The customer address.
  - The date and time of the appointment.
- Wish to be able to assign an appointment to an engineer.
- Wish to be notified when an appointment has been completed.
- Wish to find the customer for a completed survey and inform them of the results.
- Send and receive messages to/from an engineer

As a survey engineer I:

- Wish to download my appointments from the server
- Wish to see a snapshot view of my day
  - Via a dashboard view
  - Via a live tile
- Wish to be able to see my next appointment and know how long until it is scheduled to begin
- Wish to get directions to an appointment
- Call the client
- Wish to enter my notes
  - Typing
  - Speech to text
  - Inking
- Wish to take photos for a survey
  - Take photos using the camera
  - Add existing photos via drag and drop
- Wish to update a survey status
- Wish to add a survey to my calendar

- Wish to pin a survey to the start screen so I can navigate directly to it
  - Secondary tile
- Wish to launch the app via a Cortana voice command
  - Launch the app and show the next appointment
- Wish to render the survey data to a PDF and email it to a customer
- Send and receive messages to/from an engineer
- Receive notifications 30 minutes before an appointment and when I am within a certain distance (geofencing)
- I wish my profile to roam between my devices

Applications

The Solarizr App is broken into two key components – a web interface to be used by the call center operator and a Universal Windows Platform application to be used by the survey engineer.

Call Center Web App

This application will not be fully realized – instead it will be a shell that hosts a number of functions, such as appointment creation and sending notifications, that can be leveraged by the Solarizr client application. The app will be constructed using ASP.NET MVC, WebAPI, Entity Framework and SQL Server. The app will be hosted on Windows Azure and will leverage Azure Active Directory for authentication purposes – see Figure 1- Web Login Screen for an example login.
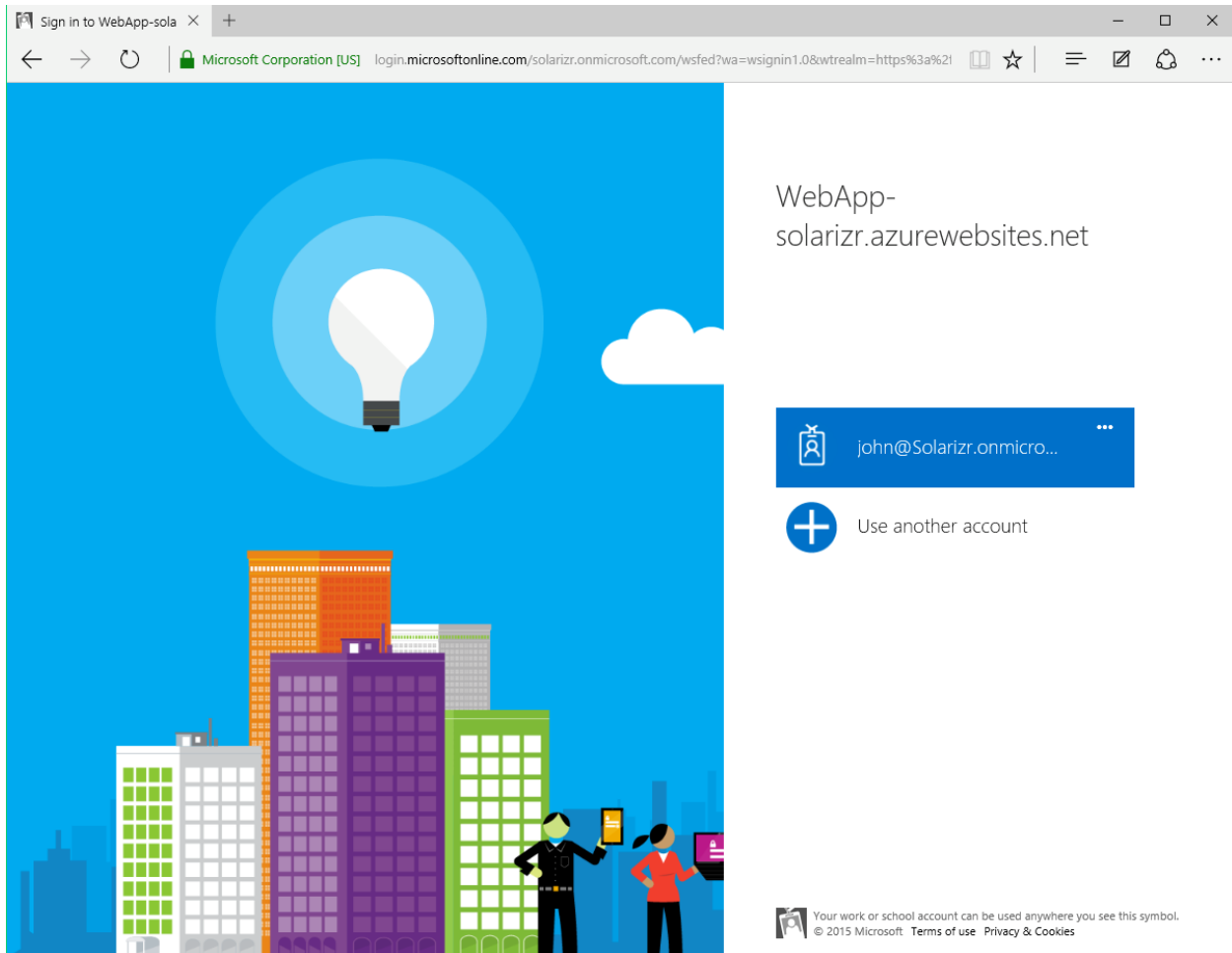
Figure 1- Web Login Screen

Web App Main Page

The main page of the web application is used to navigate to the appointment management and notification functions – see Figure 2 – Web Main Page.
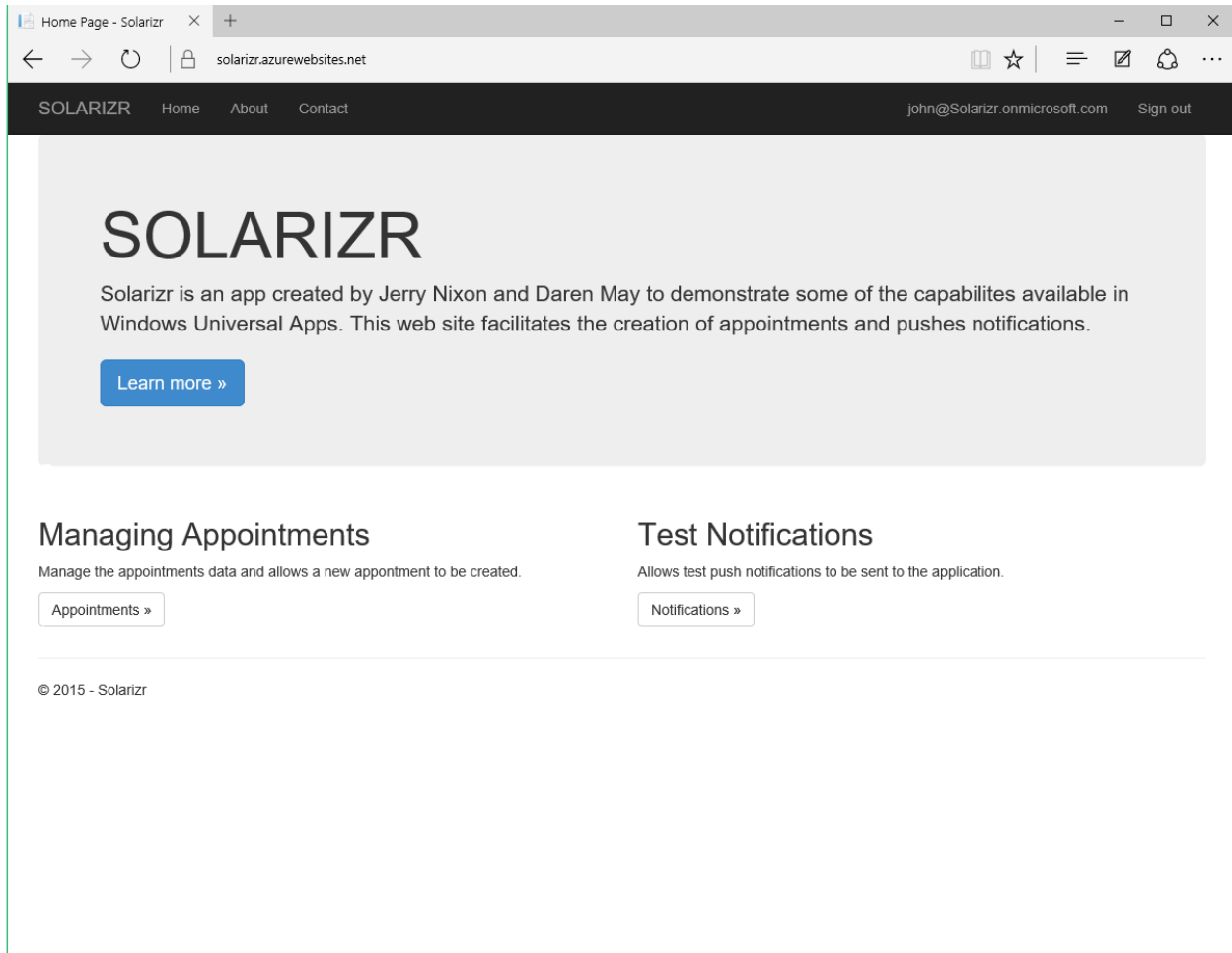
Figure 2 – Web Main Page

Appointment Management

The appointment management function consists of three pages:

- Figure 3 – Web Manage Appointments provides a list of appointments and the functions to navigate to a read-only detail view of an appointment, navigate to edit an appointment, delete an appointment or navigate to the create new appointment view.
- Figure 4 – Web Appointment Detail is a read-only view of an appointment
- Figure 5 – Web Edit Appointment is the view for editing an appointment and is virtually identical to the view for creating an appointment.

An appointment consists of the following data elements:

- ID – unique ID for the appointment
- Date – the date and time of the appointment
- Details – any text entered by the engineer
- Location – the address for the appointment
- Latitude – the latitude of the location
- Longitude – the longitude of the location
- StatusId – the current status of the appointment (Pending, Approved, Denied)
- Phone – the phone number of the customer

- Map – a pre-calculated URI for a Bing map of the location
- Name – the engineer assigned to carry out the survey

Figure 3 – Web Manage Appointments

solarizr.azurewebsites.net/Appointments/Details/5

SOLARIZR   Home   About   Contact   john@Solarizr.onmicrosoft.com   Sign out

Details

Appointment

Date	9/10/2014 3:45:39 AM
Details	
Location	550 12th Street Golden CO 80401
Latitude	39.7568714618683
Longitude	-105.218652784824
StatusId	1
Phone	303-555-1234
Map	http://dev.virtualearth.net/REST/v1/Imagery/Map/Road? key=Apk6zV2N1Keldfvh_skVQ3p36hqY4eedgEs5mlAFr3r9uD77IuOsTW0AklquxyDF&mapSize=102,102&zoomLevel=&pp=39.7568714618683,-105.21865278
Name	JOHN JOHNSON

Edit | Back to List

© 2015 - Solarizr

Figure 4 – Web Appointment Detail

Figure 5 – Web Edit Appointment

Notifications

The web app has a rudimentary capability to send a notification to the Solarizr client app - Figure 6 – Web Send Notification. The notification text would be entered into the NotificationStringField. The notification can either be broadcast to all users by leaving the UserName field blank or by entering a username – for example john@solarizr.onmicrosoft.com. The Payload field allows additional data to be sent to the app. This feature is not currently implemented.
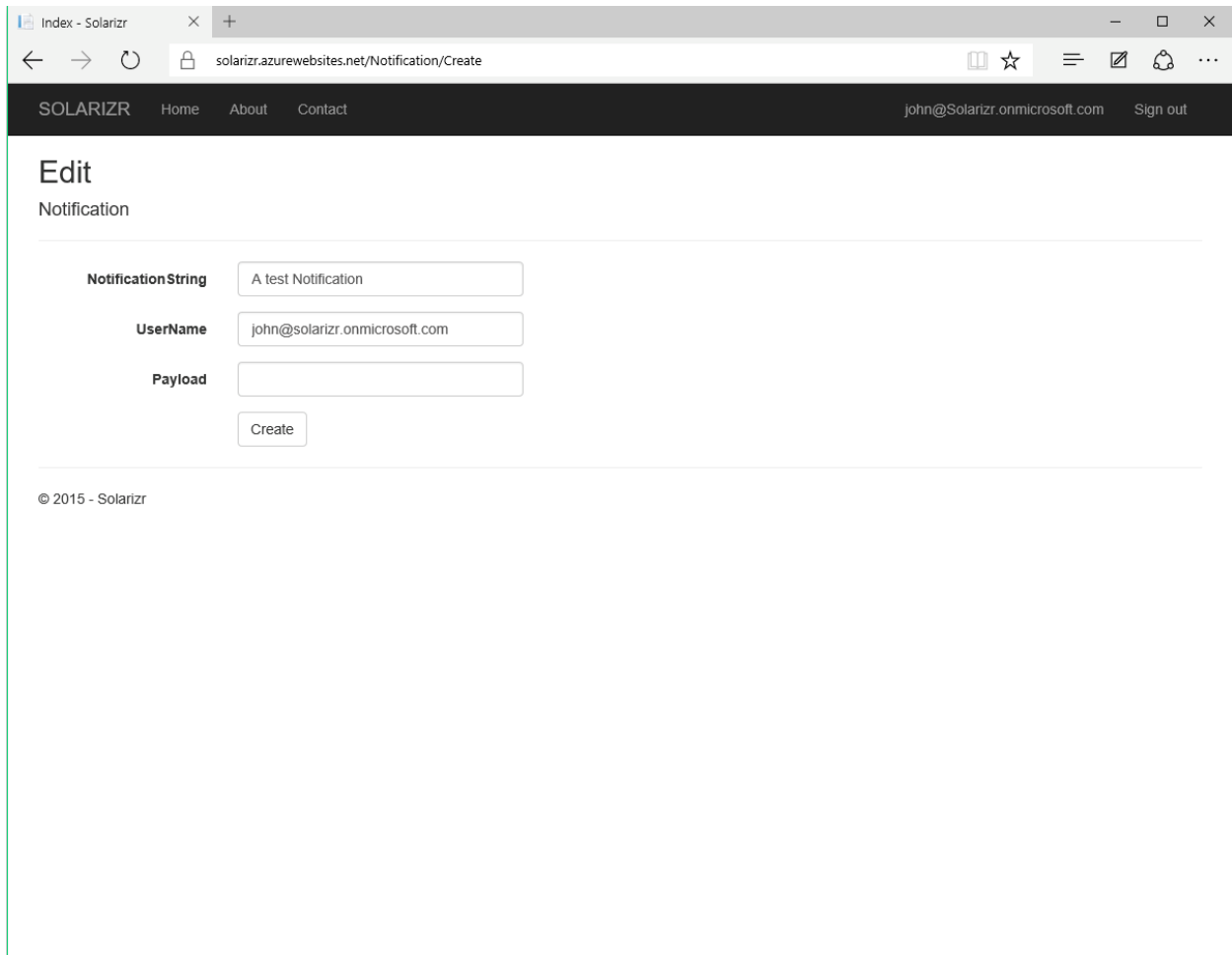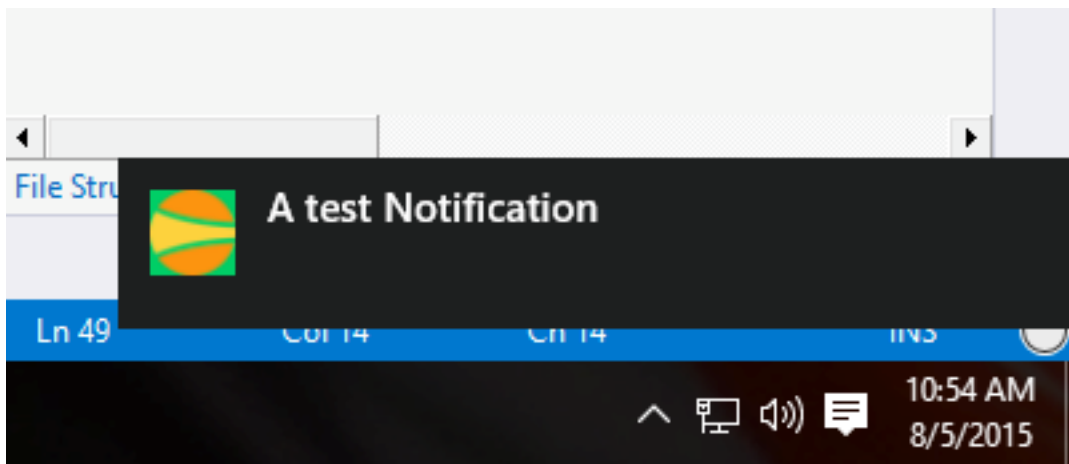
*Figure 6 – Web Send Notification*


Figure 7 – An example notification

Solarizr Client App

The Solarizr client app is built using the following technologies:

- Windows UWP (Universal Windows Platform) SDK
- SQLite database
  - This database will be used to store appointments locally before the data is uploaded to the server
- Bing Maps service
  - Used to convert addresses to Lat/Long coordinates
- Forecast.IO
  - A 3rd party weather forecast service (Bing Weather API has been shut down)
- Application Insights
  - Used to log instrumentation data

The application will leverage the "hamburger" style menu for navigation and access to settings, etc. The hamburger menu will change modes based upon a responsive UI design (mobile, narrow desktop, wider desktop and widest):
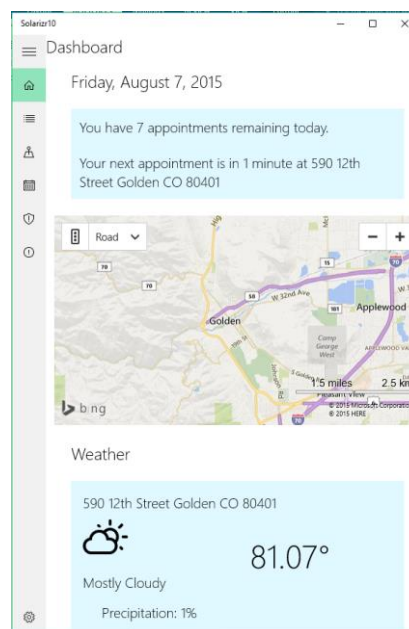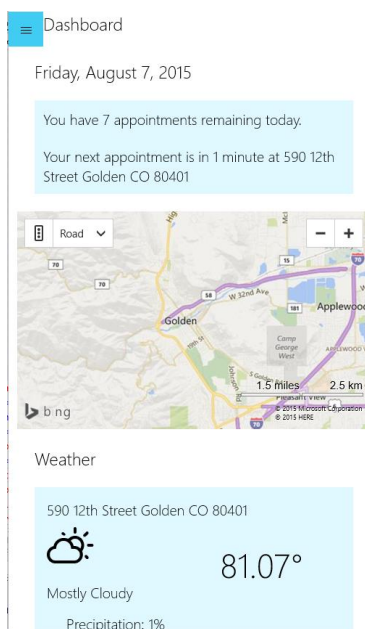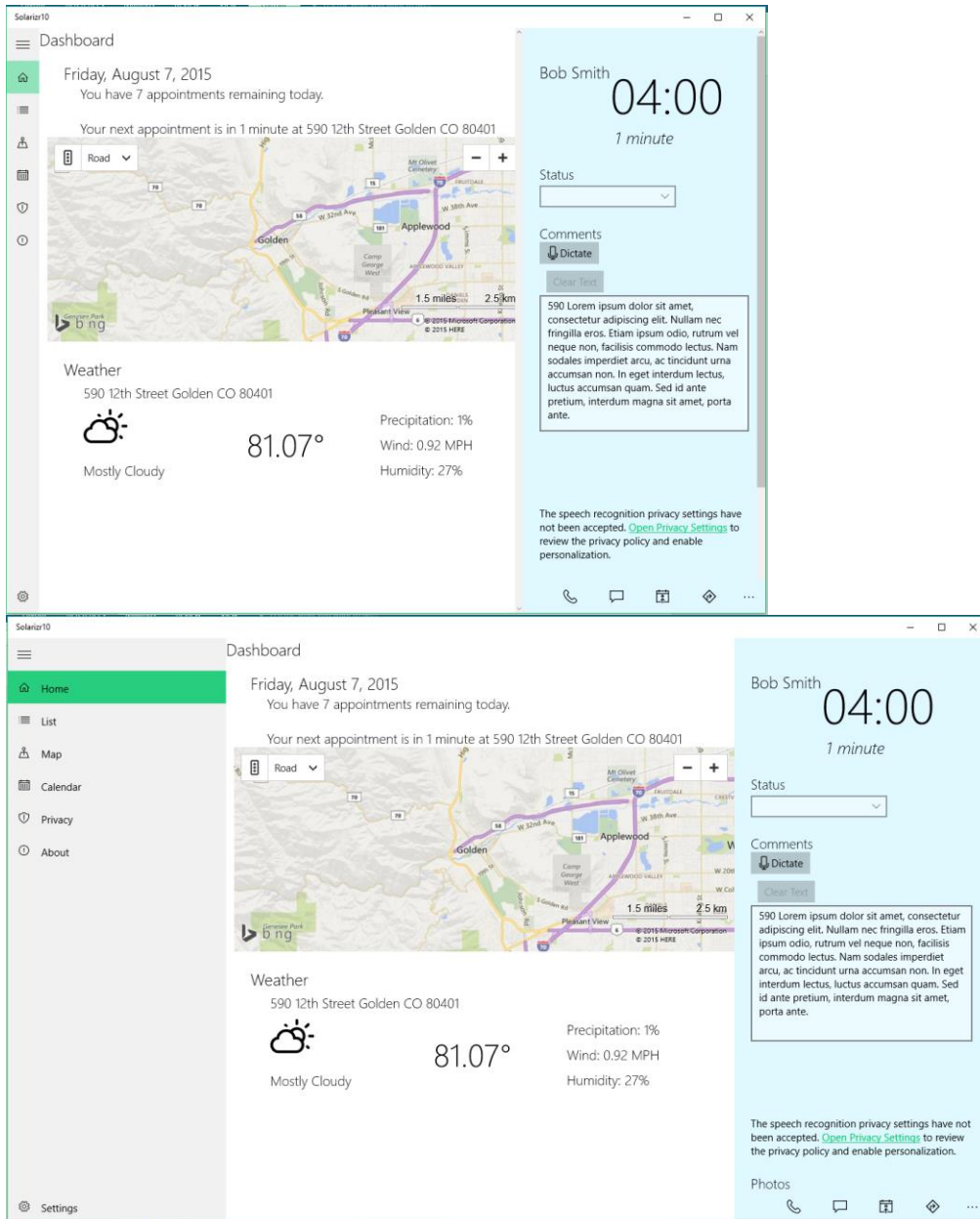
You will also note how the UI reconfigures itself to take better advantage of screen real-estate.

Client App Login Page

The Solarizr app will support user authentication. When starting the app for the first time, and any subsequent time when a valid user token is unavailable, the client login form will be displayed - Figure 8 – Client Login Page. Hitting login will then display the relevant login UI for the selected authority, such as Azure AD.
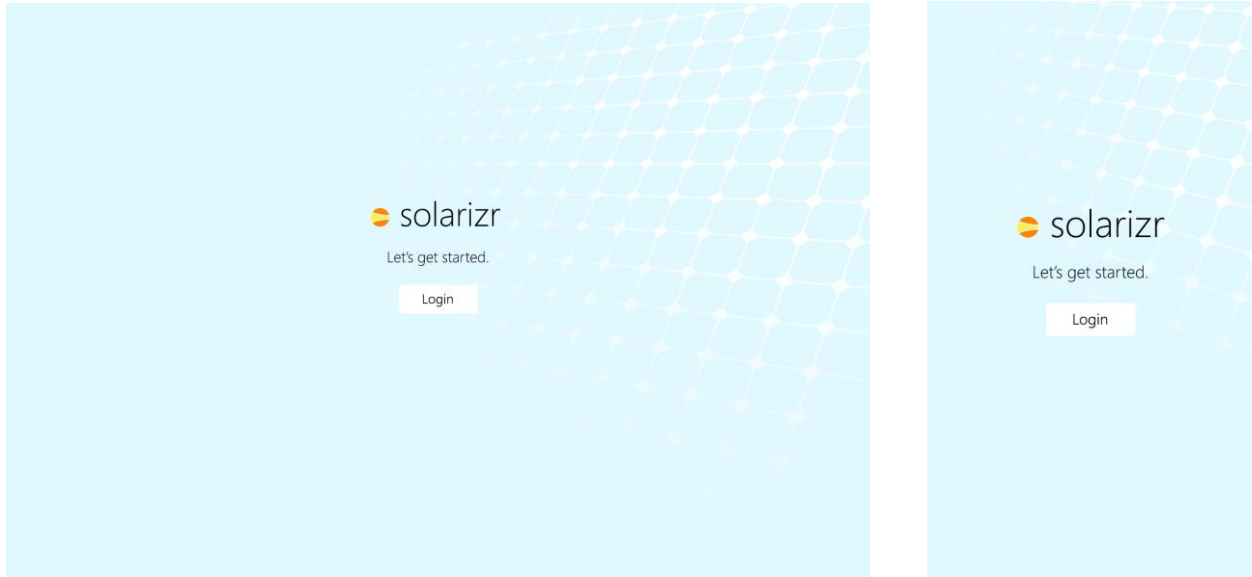
Figure 8 – Client Login Page

Client App Dashboard Page

After successful launch and login, the Dashboard page will be shown. The purpose of the Dashboard page is to provide "at-a-glance" information that supports the application function. In this instance we have elected to display the details relating to the current day:

- Date
- Number of appointments remaining
- Duration until the next appointment
- A map showing the location of the next appointment
- And where there is sufficient screen space, and editable view of the next appointment form for speedy completion of the appointment data entry without the need to navigate.

Additionally, the Dashboard page will check the server for any additional appointments that may have been created.
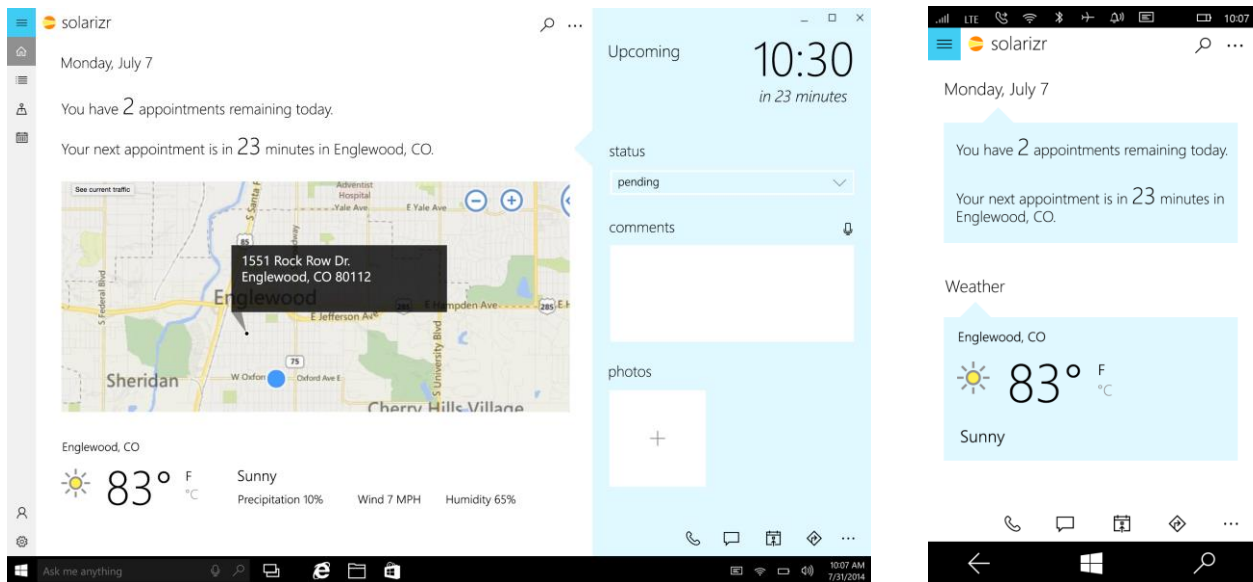
Figure 9 – Client Dashboard Page

The appointment edit form will allow the engineer to update the status of the appointment, enter comments either via a keyboard or through dictation and add photos of the location. Photos can be added via the camera, selecting existing photos via a file picker or by dragging photos onto the form.

The appointment edit form has an app bar at the bottom:
- Phone – this option will only be present if the phone capability is detected
- Chat – chat with the call center
- Calendar – add the current appointment to an external calendar application
- Directions – open an external mapping application to show directions to the appointment
- Pin (not shown) – supports pinning the appointment as a secondary tile to enable direct access
- Upload (not shown) – upon completion of an appointment survey, the data can be uploaded to the server.

Client App List Page

The list page provides an easily navigable list of the appointments on the system and on desktop systems provides a master-detail view that allows for editing the selected appointment. On phone, selecting an appointment will navigate to a second view to edit the appointment (see notes above).
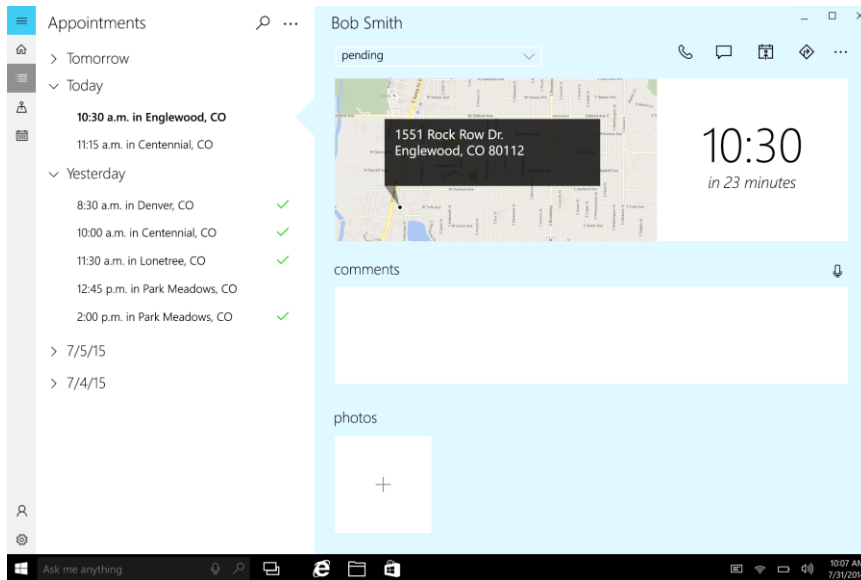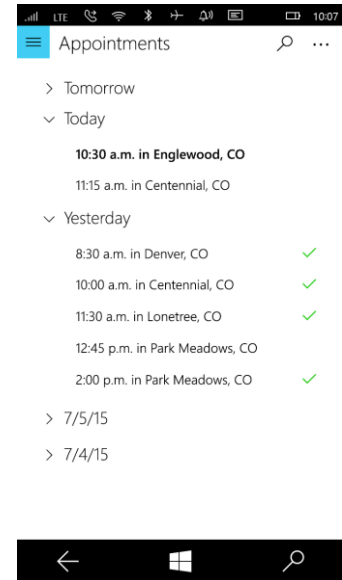
Figure 10 – Client List Page

Client App Map Page

The map page will show an alternative view that displays a filtered list of appointments on the map control. The UI will be optimized for mobile and desktop. Selecting an appointment pin will cause the appointment edit form to be displayed.
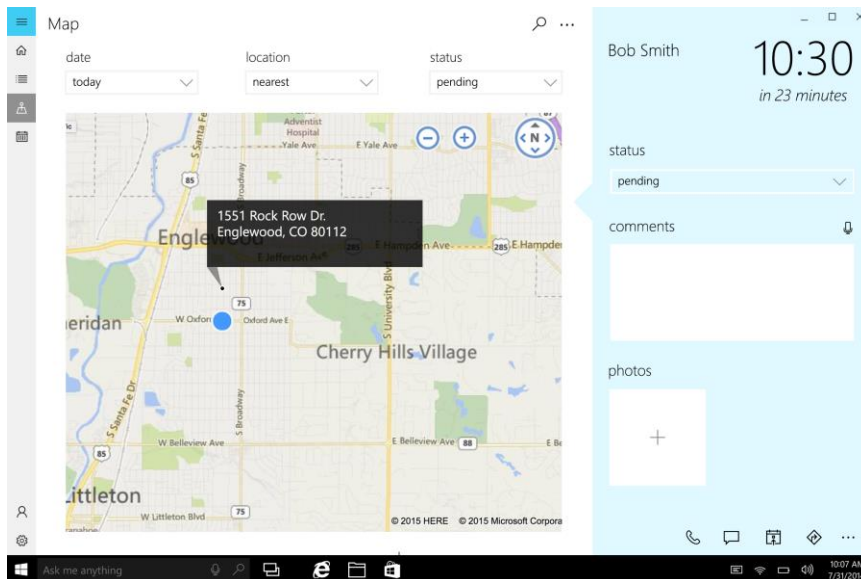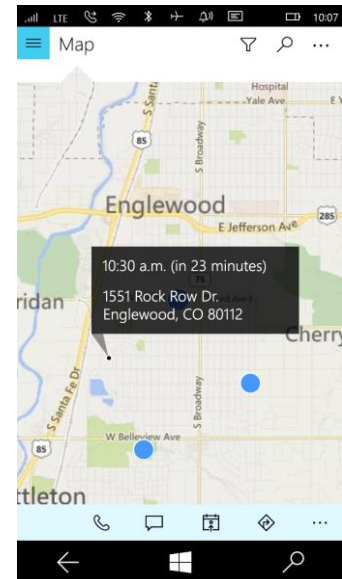

Figure 11 – Client Map Page

Client App Tiles

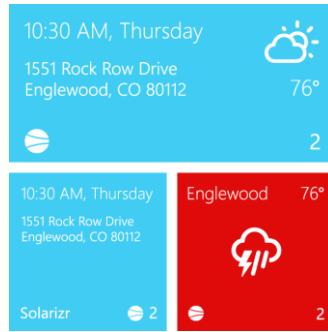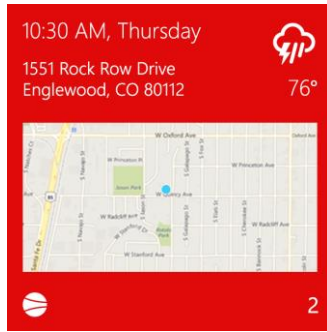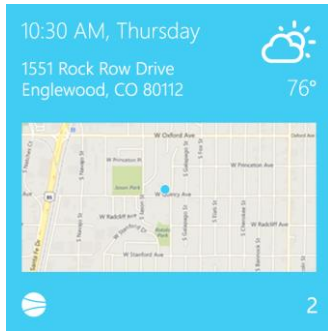The application will support the new adaptive tile format the will enable the display of rich information on the application tiles. Below are examples of the current implementation.

Figure 12 – Client App Live Tiles