GEOG 777 – Capstone in GIS Development
Project 2 Report – Nationals Park Explorer Mobile App
Kevin Palmer
11/19/2019

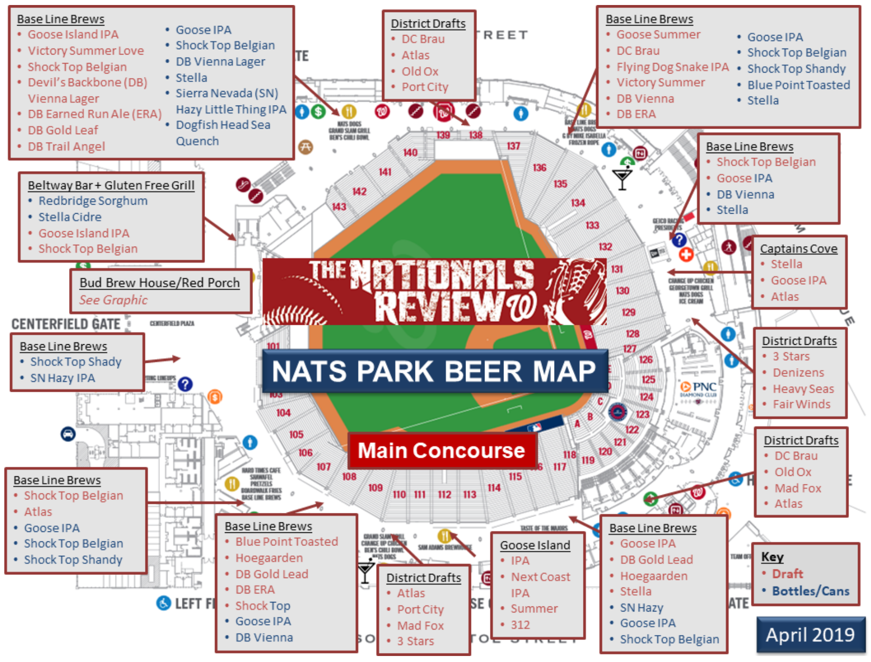
1. **A report (< 5 pages) describing the user-centered design for the application, database design including database diagrams, and implementation of the application**
2. **A video demo highlighting the application functionality**
3. **A public URL for accessing the application–useful for the peer review phase of the course (optional)**

The Problem

A park has hired me to develop a mobile-friendly application to improve the visitor experience in their park…

Nationals Park is home to the Washington Nationals major league baseball team. The park opened in 2008 and can hold upwards of 40,000 people. Driven by customer demand, the Nationals Park management has been making a significant effort in recent years to improve their concessions offerings with more options and higher quality "premium" food and drink choices for those interested in more than just standard ballpark fare. As part of that effort, Nationals Park has given particular attention to the growing interest in their offering of local DC craft beers such as DC Brau, Port City, Mad Fox, Atlas, Old Ox, etc.

Inspired by the map below created by the Nationals fan blog The Nationals Review, the Washington Nationals would like to create their own park web map application that allows users to explore the Nationals Park gameday experience and offerings on their mobile devices.


THE NATIONALS REVIEW
NATS PARK BEER MAP
Main Concourse
Base Line Brews: Goose Island IPA, Victory Summer Love, Shock Top Belgian, Devil's Backbone (DB) Vienna Lager, DB Earned Run Ale (ERA), DB Gold Leaf, DB Trail Angel, Goose IPA, Shock Top Belgian, DB Vienna Lager, Stella, Sierra Nevada (SN) Hazy Little Thing IPA, Dogfish Head Sea Quench
District Drafts: DC Brau, Atlas, Old Ox, Port City
Base Line Brews: Goose Summer, DC Brau, Flying Dog Snake IPA, Victory Summer, DB Vienna, DB ERA, Goose IPA, Shock Top Belgian, Shock Top Shandy, Blue Point Toasted, Stella
Base Line Brews: Shock Top Belgian, Goose IPA, DB Vienna, Stella
Beltway Bar + Gluten Free Grill: Redbridge Sorghum, Stella Cidre, Goose Island IPA, Shock Top Belgian
Bud Brew House/Red Porch: See Graphic
Captains Cove: Stella, Goose IPA, Atlas
CENTERFIELD GATE
Base Line Brews: Shock Top Shady, SN Hazy IPA
District Drafts: 3 Stars, Denizens, Heavy Seas, Fair Winds
District Drafts: DC Brau, Old Ox, Mad Fox, Atlas
Base Line Brews: Shock Top Belgian, Atlas, Goose IPA, Shock Top Belgian, Shock Top Shandy
Base Line Brews: Blue Point Toasted, Hoegaarden, DB Gold Lead, DB ERA, Shock Top, Goose IPA, DB Vienna
District Drafts: Atlas, Port City, Mad Fox, 3 Stars
Goose Island: IPA, Next Coast IPA, Summer, 312
Base Line Brews: Goose IPA, DB Gold Lead, Hoegaarden, Stella, SN Hazy, Goose IPA, Shock Top Belgian
Key: Draft, Bottles/Cans
April 2019
LEFT FIELD GATE
PNC

The Nationals have hired me (as a freelance web map developer) to create an application that will provide information to the user both before they arrive and while they are inside the park. As Nationals Park is located in a dense urban area where parking options are shrinking due to construction and development, they would like the app to help users get to and into the park by highlighting parking, public transportation and entrance options. Once inside the park, they would like the app to help visitors find their seats and concessions that interest them (including food, drink (particularly the craft beer selection) and merchandise). They also want to make sure visitors are able to find restrooms easily. Last, but not least, they would like to provide visitors with the option of providing feedback and rating their visitor experience through the application.

The Solution (Design)

As with GEOG777 project 1, I decided to once again implement an Esri-centric solution for this project, particularly for the server-side. While ArcGIS server hosts the application data as a service, the data itself actually resides in a PostgreSQL (10) spatial database that ArcGIS accesses through an SDE connection.

I also have experience and am most comfortable building web maps using the open-source Leaflet web mapping javascript library. In order to use both a proprietary Esri ArcGIS Server hosted services back-end with an open-source Leaflet front-end, I utilized the apply named Esri Leaflet javascript library. Esri Leaflet is an open-source javascript library that was developed by Esri specifically for this purpose.

Technology/Software Stack

Below is a list of the software components that I used in building my application:

- ArcGIS Pro and ArcMap:
  - I used ArcGIS Pro and ArcMap to collect my data from imagery and supplemented that data with information from various static maps and resources found online.
- PostgreSQL (10) and pgAdmin (4):
  - I decided to use PostgreSQL to house my data in a spatial database. I created the database using pgAdmin and then created an SDE connection to the PostgreSQL database in ArcCatalog.
- ArcGIS Server:
  - I am fortunate enough to have access to the full Esri suite of software and in order to access my PostgreSQL data via a web application in the browser, I decided to use a local installation of ArcGIS Server to host a map service containing the data. I published my data from the SDE connection in ArcMap as a single map service containing nine individual layers that could be accessed individually in the application in the browser.
  - Layers:
    - Public Trans
    - Gate
    - Merch
    - Food
    - Beer
    - Restrooms
    - Sections
    - Parking
    - ParkBoundary
- Leaflet
  - Core javascript library used for developing front-end of the application. - https://leafletjs.com/

    - I wanted my application to be a mobile-first, browser-based application, not a native mobile application. I also wanted the application to have a simple and clean interface. Leaflet was an obvious choice. I didn't show it in my demo, but the application works just as well in a desktop-based browser or in other sized devices and will adjust to suit the device the app is being viewed on.
- Esri Leaflet
    - Secondary javascript library used for connecting ArcGIS services with Leaflet - https://esri.github.io/esri-leaflet/
    - I don't know how much this javascript library gets used in the real world, but it worked out great for my purposes. Had this library not been available, I probably would have stuck with ArcGIS Server/PostgreSQL as my back-end but opted to use some combination of the ArcGIS API for Javascript or WebApp Builder to create my front-end.
- Various Leaflet/Esri Leaflet Plugins
    - Home button - Leaflet.zoomhome - https://github.com/torfsen/leaflet.zoomhome
    - GPS Locator - Leaflet.Locate - https://github.com/domoritz/leaflet-locatecontrol
    - Grouped Layer Control -Leaflet-groupedlayercontrol - https://github.com/ismyrnow/leaflet-groupedlayercontrol
    - Search Bar/Feature Layer Geocoder - Esri Leaflet Geocoder - https://github.com/Esri/esri-leaflet-geocoder
    - Star/Rating Button - L.EasyButton - https://github.com/CliffCloud/Leaflet.EasyButton
    - Level Controls - Leaflet Indoor - https://github.com/cbaines/leaflet-indoor
- Icons
    - I primarily used customized Maki icons for my layer icons. I modified a few of the SVGs in Adobe Illustrator myself to make my own (such as the family restroom icon) - https://labs.mapbox.com/maki-icons/

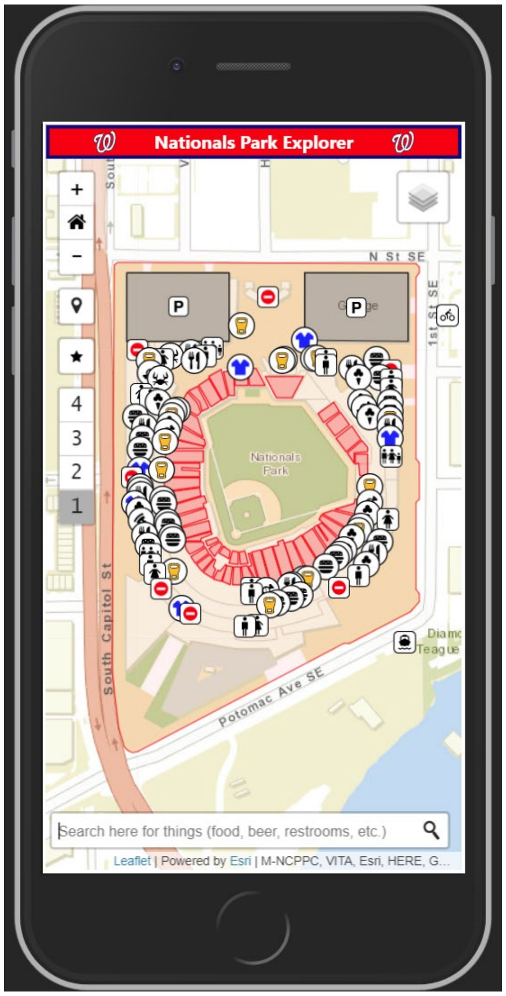
Database Design (Back-End)

- As mentioned above, the database used for this application is a PostgreSQL spatial database hooked up to ArcGIS through an SDE connection. As such, most of the database interaction was done through ArcGIS Pro or ArcMap in file geodatabase format. Each of the nine layers does share some common attribution fields across all layers. For example, the nearest section is recorded as attribution for most layers. Name and Type are also common.
- An interesting "feature" I added to the attribution on the back-end though was a "Tags" attribute field for each layer. At the very least the "Tags" field is populated with the layer name of every layer. This way if someone just searches for "Parking" for example, every feature in the parking layer results will be displayed (both lots and garages). The field can contain additional tags though. The gate layer features for example is attributed with "Gate, Entrance, Exit, Security" in the "Tags" field so any of those terms will be associated with the gate features and will display in the search results.
- [Database diagrams forthcoming]

Client Side Interface Design (Front-End)

- Features/Functions of the App
    - Dynamic layer visibility (min/max zoom level settings) "soft-functionality"
    - Grouped layer controls by category

  - Home button zooms to original extent
  - GPS locator button
  - Feature attribution popups
  - Search/Geolocator bar – this is the most important function of the app
  - User feedback/rating button (non-operational)
  - Level control (non-operational)
- (See demo video submitted along with this report for more detailed information about the app)
- The application as seen on an iPhone 6s with all layers enabled and zoomed into the park extent:


Nationals Park Explorer
N St SE
1st St SE
Nationals Park
South Capitol St
Potomac Ave SE
4
3
2
1
Search here for things (food, beer, restrooms, etc.)
Leaflet | Powered by Esri | M-NCPPC, VITA, Esri, HERE, G...

Deliverables

- This report.
- Demo video is submitted along with this report.
- Github Repo is here https://github.com/rkpalmerjr/GEOG777_Project_2_Park_Web_App.

*As with project 1…Unfortunately, I don't have a live version of this application available. I installed PostgreSQL and ArcGIS Server on my laptop and ran everything local. I do have an AWS account and eventually want to figure out how to set up a virtual server (which I'll probably do for 778), but I don't know exactly how to do that without a lot of research at this point. If you would like to see this web app live let me know and we can set up a meeting time for me to share it with you, but the demo should give you a pretty good look at what I've put together.

Potential Future Enhancements

I know that I still have some work to do to meet the user input requirement, but even without that I think I created a pretty solid mobile application with a clean, effective UX/UI that delivers on the overall stated purpose of the application, which is to "…improve the visitor experience" by being "…useful to visitors before they arrive to the park and provide information while the visitors are inside the park". As it stands, I'm mostly satisfied (and somewhat surprised) with this how well this application turned out.

That all being said, there are several enhancements that I would've liked to make to this application, in order of priority:

- Create a user experience feedback form (enabled by clicking on the star button) which allows users to rate their experience at the park and provide comments and feedback. That input information would then get populated into a PostgreSQL table.
- Figure out a way to turn on layers associated with a selected search result if not already on. Currently the app will zoom to the selected search result feature, but the associated layer will need to be turned on manually if not already on.
- Figure out a way to sort the search results in a more logical manner.
- General improvement of the popup formatting for all layers.
- Create data for the 3 upper levels (4th level is just seating) and figure out how to get the Esri Leaflet Indoor Mapping functionality working. I have a feeling this one would prove difficult to get working the way I would want it to (to control multiple layers at once).