

TIG FINAL EVALUATION REPORT

Grantee name: Ohio State Legal Services Association

TIG grant number: 15031

Approval Date: 9/28/2017

Contact person: Joshua Goodwin, Jim Daniels

Phone number: 740-773-0012

Email address: jgoodwin@oslsa.org, jdaniels@oslsa.org

I. PROJECT GOALS AND OBJECTIVES

As set forth in our Evaluation Plan, the goal of this project was as follows:

Develop a mobile application for Android and iOS powered devices called Ohio Legal Services Assistant that will increase the effectiveness and efficiency of Ohio's Legal Services Attorneys and pro bono volunteers by providing them with easy, mobile access to information and resources they need to properly advise and represent the poor.

To achieve our goal, we set the following objectives:

1. Develop and deploy an iPhone app called Ohio Legal Services Assistant (OLSA) that has the same abilities as the currently existing Ohio Legal Aid Assistant for Android devices which gives legal services and lay advocates mobile access to some rules of procedure for Ohio's Courts, commonly used help sheets, and the ability to calculate food stamps, OWF, APR, eligibility for premium tax credits and Medicaid, garnishability of income, child support, and the federal poverty level.
2. Develop a push notification system that will display important alerts and messages to users of the OLSA app without the app being open, which will help users by notifying them of relevant changes in the law, changes in how benefits are calculated, and of events of importance to advocates for the poor.
3. Develop a method that allows the OLSA app to integrate with the Pika case management system such that a legal services case handler can review their case list, client information, review case notes, and enter a time slip through the OLSA app.
4. Ensure that the apps have the Ohio Rules of Court and relevant Federal Rules of Court are available in the app and that a system is in place to ensure that the rules are always updated.
5. Develop OLSA app functionalities that allow users to easily, efficiently and effectively complete and email at least two form packets or pro se packets.
6. Develop an updateable database of services and resources for low income Ohioans that users of the OLSA app can quickly search in order to find local resources that are of benefit to their clients.

7. Conduct outreach and encourage Ohio's legal services advocates and lay advocates for the poor to start using the OLSA app, gather user feedback on the apps, and look at how the OLSA app is actually used so that the app can be improved.

There were no significant changes in the goals or objectives throughout the project.

II. EVALUATION DATA AND METHODOLOGIES

The data used in this report is the data collected pursuant to the evaluation plan. All the data discussed in the evaluation plan was collected as anticipated with the exception of the webinar information. After building the App, we determined that conducting webinars would be of no value given the ease of use of the app. Instead, we created a video for each app describing the app and what it could do. Those were uploaded to YouTube and links were made available on the store listings for each app.

III. SUMMARY OF MAJOR ACCOMPLISHMENTS, RECOMMENDATIONS, AND FUTURE STEPS

OSLSA created the Ohio Legal Services Assistant (the “App”) for both Android and iOS powered mobile devices. The App provides useful tools and information to Ohio Legal Services attorneys that enable them to more effectively and efficiently advise their clients on a number of issues.

The App includes calculators designed to help attorneys determine a client’s eligibility for various public benefits such as food stamps and cash assistance. It also calculates a client’s percentage of the federal poverty level, which is useful at clinics and for determining eligibility for various programs.

The App also has an offline version of various rules of procedure for Ohio’s state and federal courts. The offline nature of the rules is important as some of the more rural parts of the state still lack reliable data service for mobile devices. The rules are also searchable.

The App includes a local resources database as well. This database provides a list of resources useful to the low-income community sorted by county and category. It is expandable to include other counties outside of the Southeastern Ohio Legal Services community.

Two static forms are included in the App that also help determine benefits eligibility. There are also two mobile-optimized HotDocs forms created for the App to help attorneys generate various advanced directives.

Lastly, for attorneys at OSLSA, a mobile optimized version of specific Pika screens were created to make entering time in to the case management system easier on mobile devices. The changes were created in such a way that other Ohio legal services programs could easily adopt those changes in their own case management systems.

OSLSA would recommend that future versions of the App include a news feature that serves as an archive of push notifications received by the App. Additionally, consideration should be given to moving the rules and calculator files from being hardcoded in to the App to being occasionally downloaded from a website to make updating the rules and calculators easier.

IV. IN-DEPTH ANALYSIS OF ACCOMPLISHMENTS

Throughout this project, OSLSA succeeded in meeting the identified project objectives. All the code, with the exception of the JSON file needed to work with Firebase that contains private keys, available freely to anyone interested on GitHub at <https://github.com/oslsa>.

Objective 1: Develop an iOS Version of Ohio Legal Aid Assistant

We were able to create an iOS version of the App with only one deviation from the existing Android version. That one variation is relatively minor and does not change the overall functionality of the iOS App.

The iOS App is an extension of a side project called Ohio Legal Aid Assistant created by Joshua Goodwin, an OSLSA employee. The Ohio Legal Aid Assistant was more of a concept limited to Android phones than a fully formed project. The App created by this TIG grant is the fully formed mobile application.

Because it borrows heavily from the concept app, much of the Android coding was already done for the rules of procedure and the calculators. The Android work, initially, was limited to making the user interface better. The hard work in this objective was to create the iOS version of the app.

The overall design was created with a minimalist mindset, which is why there is a lot of white space in the iOS version of the App. The layout is entirely different from the Android version so that it would be familiar to iOS users. Thus, instead of using the Menu button and side drawer model used by Google, the iOS App uses the tab bar model. The color scheme is consistent with the OSLSA color scheme to reflect where the App was created.

The App was written using Apple's XCode Integrated Development Engine, which is free and only available for Mac. We had the option of writing the underlying code in either Objective C or using the Swift programming language. We chose Swift because it is Apple's official programming language and seems to be the direction more developers are heading for iOS development. In addition, it is closer to Java than Objective C, and our programmer is primarily a Java user.

Like the Android App and the concept App that predated it, the iOS App has access to the same suite of calculators as the Android App. We tested the accuracy of the iOS App's calculators by comparing the results against the corresponding calculator in the Android App, which had been previously tested for accuracy. For each calculator, we created three different scenarios of increasing complexity to test all aspects of the calculator. Fortunately, because the underlying logic was already

created, the biggest challenge in creating the iOS calculators was translating the code from Java to Swift.

Of the calculators, the poverty level calculator is the most used at 52 uses on at least Android devices from the home screen and 32 times from the calculators page. We have some reason to believe that not all iOS uses are being recorded and are hoping to figure out why. The next most commonly used is the APR calculator; however, it has only had 1 use in the last 30 days and we believe that the high number of uses is due to our efforts to fix a minor bug discovered earlier in the year. Accordingly, we believe the next most used calculator is the food stamps which has been opened 28 times. A printout of the data is attached.

In addition to the calculators, we had to create a rules of procedure portion of the iOS App. In doing so, we wanted to create something that was 1) available offline, 2) easily updateable, and 3) searchable. Ultimately, we choose to break up the rules so that each paragraph was an individual line in an iOS property list file. This allowed us to put formatting instructions in the code itself. It also meant it would be easy to identify sections that changed when the rules changed. Another advantage to this approach is that the XML files used to create the rule books in the Android App can be modified using find and replace on a few different strings and then saved as property list files in XCode. This means that rather than having to retype the rules in after creating them for the Android App, 5 minutes of finding and replacing can make the Android file usable on the iPhone. This significantly eases creation and maintenance of the rules.

One of the negatives of this process though is that occasionally it can be slow to load some of the larger rule books because property list files are slower to load on iOS devices. Additionally, this method means that all the rules are hardcoded into the app. Thus, the only way to update the rules is to update the App itself, which means resubmitting it for approval from Apple, which can take a few days to complete.

The last aspect of the concept version of the App that we created in the iOS version is the forms section. In the forms section, users are given the ability to view static forms that are useful to Ohio's legal aid attorneys (we will discuss fillable forms later). This is the only area where we were unable to port functionality from the Android App to the iOS App. Like the Android App, the iOS App can view static forms, and it is preloaded with the benefits helpsheet and the Medicaid helpsheet. However, the Android App also allows users to create links to other forms on their phones and have those documents show up in their forms list in the App. Due to the security set up of the iOS system, there was no direct analog to this functionality in iOS that we found. Accordingly, we could not include this functionality.

From our analytics data, it appears the forms section has been opened 36 times, though it is unclear whether this is counting iOS opens.

Objective 2: Push Notifications

We successfully created a push notification system without any major limitations.

We wanted to be able to provide users of the App with notifications about significant changes in the law, news alerts, and updates to the App. These notifications, called push notifications, pop up in the status bars or the iOS powered devices and provide information to the user without the user actually opening the App.

We did not want to host and manage the backend server required to a push notification system and so we opted to use a third-party back-end as a service (“BaaS”). Using the BaaS, we would not have to maintain push functionality for sending the messages but instead could focus on the content we wanted to deliver.

When we originally began thinking about the Apps, we had planned to use the BaaS application called Parse. Unfortunately, it announced in January 2016 that it was going to shutter its service in January 2017. Thus, we had to find an alternative, which led us to the use of Firebase as our BaaS. Firebase enables the capturing to app analytics, push notifications, storage, and the use of a database. It is owned and operated by Google and Google has been integrating it more and more into their developer tools. In fact, Android Studio, the official Android development platform, now comes with a Firebase tool set out of the box. Thus, we felt comfortable that using Firebase will have some guarantees of stability and longevity.

Enabling the push notification system in the Apps was relatively simple as Firebase is fairly well documented. We experienced some challenges with the upgrade to iOS 10 that broke some push notification functionality, but the Firebase community quickly helped provide a solution.

We have pushed out a few push notifications since the Apps were launched, and a copy of the launch screen are attached. When we created the App, we offered three different “tiers” or users for push notifications: legal aid, pro bono, and lay service provider. While we think this was an interesting and potentially useful concept, we have yet to see a use for this tiered system where we wanted to send a tailored message to one group and not make it available to everyone. A listing of the circumstances when push notifications will be sent is attached.

Since the official launch of the Apps in late January 2017, we have sent out three different push notifications as follows:

February 2, 2017 – updating users that the poverty level numbers had changed. 25 users received this notification.

March 23, 2017 – updating users as to updates to the static forms in the App. 30 users received this notification.

June 12, 2017 – alerting users to the survey in the App. A total of 49 users received these notifications.

Objective 3: Mobile-friendly Pika

We were able to create the mobile-friendly version of Pika that we set out to accomplish without any major limitations.

OSLSA, and all of Ohio's legal aid programs, use the open-source Pika case management system. Pika uses a version Twitter Bootstrap for its layout, which is designed for desktop computers, not mobile devices. Accordingly, it is hard to use on a mobile phone. The goal with the App was to enable a user to enter and view case information in a mobile friendly manner.

When we originally envisioned the App, we had planned on integrating with the underlying Pika SQL database and building the function directly into the App itself. This presented us with a number of technical problems. The intersection of the Pika database and the App would be a potential security risk. Additionally, hardcoding the information into the App would make expanding the usability to the other legal aid programs a challenge as each new program would have to have a special version of the App created.

Thus, wanting to minimize security risks and increase replicability, we opted to instead change Pika. We modified and created various files in the Pika program to make them mobile friendly. This meant following responsive design principles so that everything displayed nicely on the screen and was easier to navigate. Then we allow users to link to the mobile-friendly version of Pika in the App. Thus, they can open the App and quickly access the mobile version of Pika without having to remember the address. The end result is that a user can easily enter time on a case, view the case list, view a client's notes, and view the client's contact information.

The modified files are all available from OSLSA's IT department and could easily be adopted for use by any program using Pika. The Pika changes themselves were minimal. The biggest challenge is that Pika is written in PHP and it took time to learn that language and figure out where to make the changes. But, because the changes themselves, once discovered, were minor, adoption by others is easy. Future versions should spend time making the design look more aesthetically pleasing. It is functional and works well, but it is, admittedly, not the prettiest design possible.

There were various tests done to this system to ensure accuracy. Fortunately, testing was easy as it involved a side-by-side comparison of the case list in the App with the case list on the computer. Additionally, test time slips were entered on the App and the results were then viewed on the computer to ensure accuracy.

It is impossible to tell from either the app or Pika actual usage rates of the mobile version of Pika. However, we know that it is being used. One survey respondent listed it as their favorite feature of the app. Anecdotally, we have heard from several users that they enter time in Pika when out of the office via the app. Additionally, survey respondents mentioned specifically their use of the mobile Pika, appreciation of it, and wish for it to include more features.

Objective 4: Rules of Court

The rules of Court for both State and Federal Court were included in the App. This includes the State and Federal Rules of Evidence and Civil Procedure, and the Ohio Appellate Rules, Juvenile Rules, and Rules of Professional Conduct.

As mentioned, the rules are written in XML and saved as XML and property list files in the Android App and iOS App respectively. Each paragraph is broken out separately as a new line. The beginning of the line contains a number indicating how many indents the line should have to maintain formatting on the phone. This number is stripped out before displaying. Updates to the rules are made by modifying the particular line that has been amended.

One of the downsides of having everything hardcoded into the App is that changes must be resubmitted to the various stores. For Android, this is not an issue as there is only a few hours delay before updates are pushed through. For iOS, updating the rules requires review by an Apple employee before the updates go out. This can slow down updating. Future versions of the App should consider storing the rules in Firebase and having a copy of the rules persist on the device. This will allow them to be useable offline but will not require a new release of the App just to modify the rules of procedure.

The rules have become a frequently used feature of the App based upon the analytic data we have obtained. A copy of that data is attached. Anecdotally, various attorneys within OSLSA have mentioned having used the rules within the App in Court. We suspect that this will end up being the most used feature of the App in the long run.

The Rules feature of the App is one of the most used features, at least among Android users. As of June 13, the Rules were opened 43 times from the home screen, and the federal rules of evidence were the most commonly opened at 27 times.

Surprisingly, the search function is rarely used according to the usage data we have at this time.

An update schedule is attached showing when the rules get updated and who is responsible for doing the updates.

Objective 5: Fillable Forms

We successfully implemented two fillable forms in the App.

We explored two different options in adding fillable forms. The first was to build the capability natively into the Apps themselves. This would likely make the forms fillable offline and could be faster. The downside is that adding forms and updating the forms would require recoding portions of the App. Additionally, it would require the use of other 3rd party software. The other option was to host HotDocs interviews on LawHelp Interactive (“LHI”) that were mobile optimized.

We opted for the latter approach. We already had the HotDocs software, which made template development easy. Additionally, LHI was a free solution, has many years of experience, and is likely to be around for several years into the future. Additionally, it offers the ability to email the completed forms.

While LHI’s HotDocs viewer is responsively designed, the default settings created a form that was difficult to use on the phone. To make the interviews easier to use on the mobile devices, we had to change our normal design practices. Instead of using the three columns HotDocs allows and that we typically take advantage of, we left everything in one single column. This created a more vertical interview, which is easier to see on the phone. Additionally, we had the interview and resources panes hide upon opening the interview on LHI. The default behavior is to have those panes showing, which significantly reduces the screen space available for the interviews. These minor changes greatly improved the usability of the interviews on the phone.

The App simply links to the LHI interview and opens them in the phone’s default web browser. We created two different templates. One creates a living will and healthcare power of attorney and the other creates a durable financial power of attorney. These are two forms that we create a lot of in meetings with senior citizens. Additionally, these forms are ones that lay service providers also are likely to use based upon various conversations we have had over the years. Thus, we believed they would be most useful by both the legal aid community and the broader community.

The use of these forms appears to be limited at this point in time, and it is not clear whether completing forms on a mobile device will ever gain widespread acceptance by attorneys. Only the 2017 First Quarter statistics are available from

LHI regarding the use of these forms and each has only been accessed two times. The relevant portion of the statistics are attached.

Objective 6: Local Resources Database

We successfully implemented an easily updateable local resources database.

As mentioned in the section on Push Notifications, we went with Firebase as our BaaS. When we explored various BaaS's (Parse, Firebase, Quickblox, and OneSignal) for the push notification system, we did so with the hope of using the same BaaS for the local resources database. Firebase appeared to us to be the most reliable, and its database features met all of the criteria we were looking for.

The Firebase database stores information in a JSON file and uses a no SQL querying library. Thus, we had to create a JSON tree to store all of our local resource information. This meant each piece of information we wanted to display had to be broken down in to its individual parts (address, county, type, etc.) and stored with a common name.

While entities could be added directly in the Firebase web console, doing so was cumbersome. Instead, we created an HTML file that displays a local website that gathers the information and makes it easier to add entities. We then went through our list of local resources contained in Pika and added them all into the Firebase JSON file. OSLSA's pro bono coordinator, in addition to the developer, can add new entities to the database through the HTML file.

Within the App, local resources are first sorted by county. Then, within the county, the resources can be further filtered down by type. Once a resource is selected, the details of the resource are interactive. Thus, tapping on the phone number places a call, tapping the website opens the web browser, and tapping the address opens the maps feature of the phone.

There are more than 800 entities in the database and we expect it to grow over time. Maintaining the database will be a challenge, but a schedule for doing so is included in the attached Schedule of App Maintenance. Testing was accomplished by randomly choosing several entities from various counties and verifying that the information displayed was accurate.

Anecdotally, various attorneys have mentioned using the local resources database to provide information to clients. The App's developer uses it regularly for this purpose. Additionally, usage analytics show that it is one of the most commonly used features, at least among Android devices. As of June 30, the local resources had been opened 75 times, making it the most used feature of the App for which Android usage is tracked.

Objective 7: Outreach and Reflection

We conducted various outreach efforts and gathered feedback. The only significant change to the evaluation plan we made was not conducting webinars. Given the relative simplicity of the App, we determined that webinars would be too short to be meaningful and likely would have attendance too low to be useful.

In order to find users for the Apps, we engaged in the following outreach efforts:

1. January 17, 2017 – email to early tech adopters with OSLSA and directors for them identify other early adopters with the goal being to work out minor bugs that would prevent full adoption by everyone later.
2. January 24, 2017 – encourage pro bono attorneys at the Fairfield Pro Bono Clinic to use App.
3. January 30, 2017 – email to staff regarding the address for the mobile version of Pika.
4. February 28, 2017 - encourage pro bono attorneys at the Fairfield Pro Bono Clinic to use App.
5. March 28, 2017 - encourage pro bono attorneys at the Fairfield Pro Bono Clinic to use App.
6. April 25, 2017 - encourage pro bono attorneys at the Fairfield Pro Bono Clinic to use App.
7. May 19, 2017 – email to all OSLSA staff about the App.
8. May 19, 2017 – presentation to all OSLSA staff about the App at our programwide retreat.
9. May 23, 2017 - encourage pro bono attorneys at the Fairfield Pro Bono Clinic to use App.
10. May 23, 2017 – email from OSLSA pro bono coordinator to pro bono volunteers about the App.
11. June 7, 2017 – email to all legal services advocates in Ohio about the App.
12. June 7, 2017 – email from Mike Walters at Pro Seniors to all Pro Seniors staff.
13. June 8, 2017 – there was a post on the Southeastern Ohio Legal Services Facebook page about the App.
14. June 13, 2017 – the head of the Fairfield County Housing Coalition shared news of the App via email across the Fairfield County and Licking County social service listservs.
15. June 27, 2017 – the app was recommended during the taping of a self-study CLE on brief advice clinic volunteering developed by the Ohio State Bar Association and the Ohio Legal Assistance Foundation.

As of June 30, 2017, the Android version was installed on 21 devices and the iOS version had been installed 67 times. The iTunes page for the app had 1570 impressions and the page had been viewed 73 times. A similar statistic for impressions is unavailable for the Android version. Statistic reports supporting these numbers from Google and Apple are attached. It is clear that more attention should be given the iOS App as it appears likely to always have more users.

We also created two YouTube videos, one for each version of the App and posted them on YouTube. Links to the videos were posted on the iTunes and Google Store pages for the Apps. Unfortunately, there has been incredibly low viewership of the videos. The iOS video has been viewed 10 times and the Android video has only been viewed 3 times. In hindsight, efforts on the videos could have been put to better use given the low viewership. In discussing the video concept with various attorneys in OSLSA, none of them showed any interest in taking time to watch the videos. Perhaps videos for relatively simple mobile apps are not useful.

On June 11 and 12, the Apps were updated to link to a survey about the App and a push notification was sent to all users. Additionally, we asked all OSLSA advocates who downloaded the app to complete the survey. A copy of the survey and its responses are attached.

We received a total of 28 responses to the survey. 53% of the responses came from iOS users and 46.4% were from Android users. All but 1 of the Android users rated the appearance of the app at 4 or higher on a scaled of 1 to 5. iOS users rated the app's appearance lower, at 3 or 4, with one 2. The comments provided on appearance mention that the app is "utilitarian" and "boring", though it was also noted that this suited the app's purpose.

In general, the respondents felt the app was easy to use with 25 out of 27 responses rating ease of use 4 or higher out of 5. The comments to this question note that the app is intuitive and straightforward.

When asked about problems experienced with the app, iOS users reported the app crashing occasionally, often when opening for the first time. We are in the process of trying to track down that bug, but because it is intermittent, we are having troubles identifying it. Android users reported issues connecting to Pika. This is a relatively recent issue, and we believe it might be related to a recent update to Google Play Services, and we hope to identify and resolve this issue in the next week or two.

Respondents identified a few items for expansion. They want more updates to the app, though it is unclear what this refers to; the apps are updated as the rules and calculators change. There is also a desire to expand the mobile accessibility of Pika to include case name search and other features, which we agree would be useful.

Additional handouts were also suggested as well as the ability to print screens via bluetooth. Some of these items are included in our proposed next steps below.

Of the portions of the App for which deeper analytic hooks are available, we know that the local resources are overall the most frequently used. From February 1, 2017 through June 30, 2017, county information was accessed 75 times. The federal poverty level was calculated 52 times from the home page of the App. Lastly, the rules of court were accessed 49 times from the home page of the app, making that the 3rd most popular use of the App. Everything else was accessed fewer times during that period.

How the App Improves Service Delivery

In discussing the App with advocates in OSLSA, we have seen how it has been used.

At clinics sponsored by OSLSA, some advocates are using it to quickly check an individual's financial eligibility based upon the federal poverty level. Additionally, at these brief advice clinics, it is helpful to be able to quickly locate the resources available in the county that can help clients and provide them with the appropriate contact information.

One advocate specifically mentioned using the App in court to quickly look up one of the rules of evidence and showed the rule to the judge via the App. We anticipate hearing other uses over time.

One advocate has reported using it regularly to enter time slips via the mobile optimized version of Pika when he is out of the office. Another, in a survey response, stated "the ability to access and enter time while away has been extremely helpful in keeping up with time keeping".

V. INFORMATION FOR MULTIPLE YEAR OR MULTIPLE PROJECTS

Not applicable.

VI. FACTORS AFFECTING PROJECT ACCOPMLISHMENTS

There were a couple of difference challenges that made completion of the project more challenging than anticipated.

The first centered around the push notification system. As mentioned, the project was originally conceived with the idea of using Parse as the backend. Parse had been purchased by Facebook and we had anticipated that it would likely be around in some form. It was also very popular in the developer community. Lastly, the person developing the App had experience using Parse from other projects. Thus, all planning had focused on it as an option.

When it was announced in early 2016 that Parse was shuttering, we were left in a lurch and had to do research we had not expected to do. Additionally, we had to learn how to incorporate Firebase. This was relatively easy for the Android App, due in large part to the fact that Google owns both Android and Firebase. The documentation for the iOS version of Parse was both underdocumented, and in some cases, simply wrong. It took quite a bit of searching to figure out how to implement it in the first instance.

Lastly, once we had it working, Apple released iOS 10 which broke all functionality for Firebase push notifications. There was no documentation for some time and it left many developers scrambling to fix. Ultimately, a fix was discovered, but the push notification system was a bigger challenge than anticipated.

The other major challenge was the mobile optimized Pika pages. While Pika is open source and the code is available on GitHub, the developer for this project was not very familiar with the language Pika is written in. Additionally, the inline documentation on GitHub for PikaOCM may be helpful to those who are intimately familiar with Pika but less so for those wanting to make occasional tweaks. As a result, it took a lot of effort and attempts to make the necessary tweaks relative to the number of changes that were ultimately needed.

Ultimately, by going the route of changing Pika instead of building the Pika integration into the App will be useful to other Pika users in both Ohio and across the country as they can look at our modifications and modify their own instances of Pika accordingly.

VII. STRATEGIES TO ADDRESS MAJOR CHALLENGES

As mentioned, the major challenges in this project centered around modifying the Pika files and the push notification system.

With respect to modifications to Pika, we reached out to another developer familiar with Pika regarding what we wanted to do. While he was unable for various reason to make the changes we needed made, he was helpful in providing a general outline for what changes would need made. When we would get stuck, he would provide us with guidance about what we would need to look for to be changed.

The problems with the push notification were incredibly frustrating. Unfortunately, there were no good strategies aside from constant monitoring of the Firebase developer forums for a few days until someone discovered the solution.

In general, the developer community is incredibly friendly and wants to help. We regularly used Stackoverflow.com to answer development questions. Whenever we came to a point where the answer was not obvious, consulting Stackoverflow often provided the solution.

VIII. MAJOR LESSONS AND RECOMMENDATIONS

1. Important Lessons We Learned

One of the lessons we learned came very early on when Parse announced it was shuttering. Intuitively, we knew there were risks in relying on a 3rd party BaaS, especially a free offering. Yet, the news of Parse closing came as a surprise to not only us but a lot of developers.

We did not change the overall approach of relying on a 3rd party BaaS, but when we were researching, this issue with Parse was always in our mind. Instead, we looked for a BaaS that was tied to a major partner, like Parse was with Facebook, but also one where the major partner was showing signs of integrating it into the major partner's activities. Facebook had never made Parse a part of Facebook from what we could tell. Google, on the other hand, was incorporating Firebase in to its core products, which helped ease our minds that it would be around in the future. This insight about things to consider in looking for a BaaS was an important lesson, one that we fortunately had time to learn before coding the push notification portions of the App.

The other major lesson is that building and maintaining an App built in native code for both iOS and Android is hard. It really is building and maintaining two separate programs. The advantage is that it runs faster than using something written in JavaScript or HTML5 and then put in native wrappers for iOS and Android, but maintenance is messy and takes twice as long. Ultimately, we believe that the snappier performance of native code leads to a better user experience, but it is a tradeoff.

2. Recommendations for Other Legal Services Programs

Mobile app development can be time consuming and difficult. Other states can start from the code base we have made available on GitHub, but it will require significant tweaking of the calculators. The Local Resources tab would be the easiest to replicate as would the push notifications. Changing the rules from Ohio to another state would be relatively easy, though it is time consuming to convert the rules in to the proper XML format.

Before investing in mobile apps, we would recommend at least considering whether a WebApp would be a better option. If internet service is readily available in the area where the device will always be connected, Web-based apps might be a better solution. They might be slower than a native app, but they would be significantly easier to build and maintain. For OSLSA, especially in the rural areas, internet connectivity can be spotty at best in locations. Therefore, native apps were required to meet our goals.

3. Recommendation for Further Development

In the next major version of the App, we are considering a couple of additions.

A. A News Stream

As we push out notifications, tapping on the notification takes you to the App's homepage. However, it would be helpful if it would instead take you to a page that had an ongoing archive of prior push notifications as well as the current notification. Essentially, it would include a news stream of changes and updates to the App.

B. Searchable Local Resources

Currently, the local resources can be filtered by county and category. It would be helpful to be able to type in the name of an entity and search that way. The example that comes to mind is that the user knows the name of the entity and wants to jump right to it without going through the filters.

C. A Built-In Method for Suggesting Local Resource Additions/Edits

Currently, if a user wants to suggest edits to a local resource or add a new resource, they have to exit the App and send an email to apps@oslsa.org. It would be better if there was a page in the App where they could make those suggestions directly.

D. Persistent Local Resource Data

Currently, the local resources are only available if there is an internet connection, a problem identified by one survey respondent. The next version should gather the data the first time the app is loaded and persist it on the device so that it is available even when there is no internet connection and then update accordingly.

E. Move Calculator Data to Persistent Data on Firebase

It would make maintenance easier to move the underlying calculator data to Firebase and persist the data in the App. This way they could still be used offline but updating the data would not require a new release of the App.