Help SGF: App Completion Proposal and Outline

Myke Bates, Annie Busch, Luke Herman, and Hannah Kassing

8 January 2020

Brief Synopsis

Annie Busch created an app and website, called Helf SGF, with the original goal of creating a one-stop shop for homeless shelters, food pantries, and other areas of refuge around Springfield. She hired four coders to create this, however, it was never finished. Myke Bates is currently developing the app who was originally hired as a coder, and is now working as a volunteer. While Myke is continuing to progress with the development, further assistance is needed in order to launch the necessary features and create the desired impact in Springfield within a timely manner.

The majority of the website, at https://app.helpsgf.com/, and app is in place, but we are looking to finish them and add a few more features in a two-phase process.

Phase 1: Updates to Structure

Finish the current structure of the website:

1) Adding the "login" and "create an account" buttons on the public website (*Myke has offered to help with this)
2) Ensuring all information is editable via a web login for all nonprofits.
3) Creating the ability for homeless shelters to login through the app and update their current capacity
    a) With this, a distinguishing feature must be created that makes this only an editable feature for homeless shelters

Phase 2: Messaging

Create a messaging feature that allows nonprofits and donors to be connected.

1) Users will be able to post goods that are available to be picked up
2) Notifications will be sent out to applicable individuals with the ability to claim the items
3) Messaging will be enabled between users in order to discuss logistics

Help SGF Architecture Notes

There are a lot of pieces involved with the Help SGF setup; these are the high level details regarding the main components.

Content Admin Website/API

All of the content that the web and mobile app pull from comes from the content administration site(https://helpsgf.com/umbraco) which is powered by Umbraco V7(a .NET CMS).

The server that this runs on is managed by The Library Center and VPN credentials are needed to access the server in order to pull changes down from the Git repo and make any other changes needed at the server level.

Documentation can be found here https://our.umbraco.com/Documentation/Getting-Started/Backoffice/index-v7

NOTE: Version 8 is the newest and latest version of Umbraco, but Help SGF is running on V7. The documentation pages will often default to V8.

Mobile App

https://github.com/sgf-web-devs/Help-SGF-App

The mobile app is the main point in which users will interact with the Help SGF service. It is built on Xamarin.Forms which leverages the .NET platform and uses a single code base(C#) to cross-compile to iOS and Android.

Xamarin documentation can be found here

https://docs.microsoft.com/en-us/xamarin/

NOTE: There is a difference between Xamarin.iOS/Android and Xamarin.Forms. Xamarin.Forms mostly relies on a single codebase for UI construction while Xamarin.iOS/Android require separate implementations for each platform. Being as the UI for this app is relatively simple we have chosen to use Xamarin.Forms for simplicity.

Web App

https://app.helpsgf.com/

https://github.com/sgf-web-devs/Help-SGF-Web

The web app exists to serve as a place to go when you don't have access to the mobile app or would just simply like to pull up on any computer.

For the most part it mirrors the functionality and design of the mobile app. Any differences that do exist in functionality are on a to-do list to straighten out. For example, when filtering results - the web app does not have the sub-category filtering like the mobile app does. This is on a list to be addressed.

The web app is built with .NET Core Razor Pages

https://docs.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-3.1&tabs=visual-studio

As well as various other web tools (Webpack, Sass, Etc)

Search Service

All of the above segments utilize Algolia https://www.algolia.com/

Algolia is what powers the search results used on the mobile and web app. When content is indexed on the Content Admin site, said data is then pushed to Algolia for must faster and more robust search results.

Phase I: Core Structure and Update Feature - Time Estimation (15-30 hours)

The goal of the Core Structure and Update Feature is first to complete the base of the app and website. The second is to enable homeless shelters to update a few pieces of information themselves through the app, and the rest through a web login.

Notes from Myke:

- An update to the Umbraco portion would need to be made so the concept of an "available unit" is in place. This database houses more than just homeless shelters, and there is no distinguishing between what is a homeless shelter and what is a food provider, clothing provider, etc. This feature would therefore need to be made available only to homeless shelters, which will take some adjusting since it is not a direct fit with the current structure of the system.

Example: Jim owns a homeless shelter. By entering the app with his organization's login, Jim is able to mark his current capacity as either "Open," "<25 beds remaining," "<10 beds remaining," "<5 beds remaining," "Full." To update further information (such as the rules for his shelter), he utilizes his computer and is able to change it through the HelpSGF website with the same login.

*Myke will be creating the ability to login through the website. The core feature needing to be developed for this phase is the ability to login through the app, and to update the "Current Capacity" feature.

Technical Aspects:

- A login screen
- Authentication check and caching of identity
- App UI updated with a menu option to get to the login screen
- Account information screen with one updatable option (available only to homeless shelters)

Phase II: Messaging Feature - Time Estimation (40-60 hours)

The goal for the messaging feature is for authenticated users to be able to send message within the app with the following pieces of information:

- Title
- Message
- Image (optional)
- Audience (example: food, emergency, ride, clothing, shelter)

The message will then send a push notification for the user audience subscribed to the given segment.

Example: Mary runs a restaurant and is plugged into the Help SGF system and has the app on her phone. At the end of the day she has left over food that she wants to donate to the Help SGF crew. She opens the app and creates a new message. She supplies a Title "Food available for pickup" along with a Message "I have several loaves of bread that can be picked up between now and 10PM". If she wants to she can snap a picture of it, and then lastly she selects the "food" Audience for the recipient and hits send.

At which point, all of the Help SGF members that have previously been configured by Annie and team to be assigned to the "food" audience will receive a push notification on their phone. When clicked on this will take them to a screen to see the provided details from Mary along with her contact information that exist in the system. There is a button to press to claim responsibility for the message. Once someone presses this to confirm that they are handling it, a status message will appear on the details screen. This way anyone else who is in the audience will see the status of the message in real-time.

Technical Aspects:

Most of the data structure for this feature already exists in the Umbraco system including stubbed out example API endpoints for posting messages and pulling message content.

Further refinements would be needed to the Umbraco side to harden the existing proof of concept. On the mobile side, the process has not been started, so this would need to be done from scratch. The following are the main things that would need implemented.

- Message list screen to see list of all messages
- Message creation screen
- Will need to utilize Xam.Plugin.Media plugin or something similar to access the camera

- Data will send to the Help SGF API (Umbraco)
- Push notification subscription via OneSignal
- Setting this up would require involvement from Myke. As long as the code is setup to have some kind of entry point from a global event, then the specifics of the push notification should be able to be dealt with out-of-band with the rest of the work
- Real-time notifications (status updated of a message) will be handled via PubNub. A free account can be setup and utilized for testing. Proper account keys will be leveraged after development is complete