1.0 Working Prototype

The PQVP DS-AD Application allows California residents to establish and manage their profile and receive emergency and non-emergency notifications via email, Short Message Service (SMS), and/or push notification based on the location and contact information provided in their profile and/or the geo-location of their cell phone (if they have opted in for this service). In addition, the application provides the authorized administrative users with the ability to publish notifications and track, analyze and visualize related data.

The URL to the IFG Prototype is: http://adpq.ifgcloud.com

This readme.md file contains a description of the project. It includes a link to the installation instructions contained in the installation.md file. Documents pertaining to the development process, such as meeting notes and other artifacts, have been published in the IFG Digital Services Playbook directory within the project's GitHub repository.

IFG's working prototype source code, design assets, and all associated documentation is available online and accessible in a GitHub repository. The URL is: https://github.com/theifishgroup/chhs-prototype

2.0 Technical Approach-Description / Narrative

Since its formation in 2008, The iFish Group (IFG) has had a long and successful history in developing and delivering Information Technology solutions for our State clients. For this prototype, the IFG team utilized a tailored agile development methodology which was centered around User-Centered Design (UCD). In addition, we utilized the U.S. Digital Services Playbook plays (from February, 2017), which also include information regarding tailoring and customization for IFG projects. This information is contained within the README.md file located in the GitHub repository. In addition to the README.md file, the GitHub repository also includes the complete set of plays, the IFG Digital Services Playbook, prototype link, and tutorials instructing users on how to operate and navigate the prototype.

The IFG teams' goals were to build a collaborative team comprised of business, project, technical and development resources to produce quality working software designed and built through continuous input and feedback throughout the development of the prototype. To accomplish this, we built 13 plays utilizing the U.S. Digital Services Playbook as our template:

- Play 1: Understand what people need [Play 1]
- Play 2: Address the whole experience, from start to finish [Play 2]
- Play 3: Make it simple and intuitive [Play 3]
- Play 4: Build the service using agile and iterative practices [Play 4]
- Play 5: Structure budgets and contracts to support delivery [Play 5]
- Play 6: Assign one leader and hold that person accountable [Play 6]
- Play 7: Bring in experienced teams [Play 7]
- Play 8: Choose a modern technology stack [Play 8]
- Play 9: Deploy in a flexible hosting environment [Play 9]
- Play 10: Automate testing and deployments [Play 10]
- Play 11: Manage security and privacy through reusable processes [Play 11]
- Play 12: Use data to drive decisions [Play 12]
- Play 13: Default to open [Play 13]

Throughout each play, we built a checklist to be followed, and key questions to be answered, within the play. As part of our standard agile development process we performed the following actions to support the plays:

1. **Requirements Gathering**
    a. The first step was to obtain the "customer" requirements through a Joint Application Design (JAD) session. We achieved this by asking questions about pain points in current process or system and how the process can be automated or improved in the system.
    b. Then we developed a Wire Frame (low-fidelity mockup) of the desired screen. During meetings with the "proposed" customer, the wire frames were revised to meet the desired requirements.
    c. The wire frames were then approved for development to start.

2. **Story Development and sizing**
    a. IFG developed User Stories around meeting the requirements of the wire frame. They were used to describe system behavior. In the initial design process, we identified personas (representing generic types of potential user) and scenarios (representing potential use cases of the application). For each persona and scenario, we generated a user story that complies with our agile development process. Each user story includes a testable description of the distinct, granular, and verifiable activities which the system is expected to perform and includes detailed acceptance criteria. A customer representative (Product Owner) approved these User Stories as part of the iteration review and grooming meetings.
    b. Once a story was developed, it was then sized. The Developers developed the estimate regarding the relative difficulty of the story. Once the team had an established a velocity (how many story points the team could produce in each iteration or sprint), then we established an estimate of how long it will take to complete the given backlog of stories.
    c. The backlog of stories was then prioritized in each iteration.
    d. The number of velocity points that a team can produce in a given iteration determines the number of iterations required to complete the entire backlog.

3. **Iterative Development and Quality Assurance (QA)**
    a. After the story backlog was prioritized in each iteration, the Development team then worked on each story.
    b. Each team member completed a story in order of priority and delivered it to the story requestor (typically the Business Analyst).
    c. If the work met the acceptance criteria of the story, then it was accepted and completed. If not, it was rejected for correction and the delivery cycle was restarted for that story.

4. **Testing**

Testing was performed with both manual and automated testing routines. The testing criteria, scenarios and associated information and artifacts are located in the GitHub repository.

5. **Deploy the solution**

Throughout the prototype development, we visually captured each play, and posted the pictorial evidence for each whiteboard session for each play. These captures are located in the GitHub repository as JPG files with the corresponding play number.

3.0 Technical Approach-README.md File

The response to these requirements are located here: readme.md