Agile Development Pre-Qualified (ADPQ) Vendor Pool

Request for Information (RFI) #75001

xFusion Technologies Prototype – SafeKids

SafeKids Overview

SafeKids is a Child Welfare Digital Services (CWDS) platform with the goal to provide Children's Residential Licensing, Case Management, Resource Management, Court Process, Eligibility and Financial Management functions electronically.

The SafeKids prototype phase includes the following business functions for the Biological/Adoption Parents:

- Sign-In;
- User Registration;
- Parent Profile Management;
- Child(ren) profile Management;
- Foster Care Facility Search; and
- Messaging to Case Worker and Foster Parents through Private Mailbox.

Assumptions:

1. The "*Parent of Foster Kids*" in the Attachment B, Section 1 Working Prototype means the Biological/Adoption Parent and is the *only* User/Actor for the SafeKids prototype phase functional scope.
2. The Biological/Adoption Parent is already known to the system by their First Name, Last Name and EMail ID. For online registration, the Biological/Adoption Parent need to provide First Name, Last Name and EMail ID that should match with the existing record. The prototype doesn't perform the validation but assume that the First Name, Last Name and EMail ID are already registered with the system.
3. Biological Parent/Adoption Parent may need to provide various proof for the relationship with the child including genogram etc. For the prototype we are only asking the Parent to provide Child's Birth Certificate or Adoption Certificate.
4. Biological/Adoption Parent being the *only* User/Actor can send message to the Case Worker or to the Foster Parents but won't receive message from them in this prototype scope.
5. The Foster Care Facility Search will return Facility data only if searched using a ZIP code supported by the Community Care Licensing - Foster Family Agency Locations List at the following location: https://chhs.data.ca.gov/Facilities-and-Services/Community-Care-Licensing-Foster-Family-Agency-Loca/v9bn-m9p9

Please refer to the following deliverable for detailed business requirements in terms of User Stories.

- *ADPQ-SafeKids-UserStories*

Architecture Approach and Best Practices

Below are the key architecture approach/best practices used for the SafeKids application. For details please refer to the *ADPQ-SafeKids-ArchitectureDocument, Section 2.1*.

1. N-Tier Distributed Architecture
2. Micro-service based Architecture, RESTFul API
3. Technology Agnostic Application Layer
4. Human Centric Design and Responsive UX/UI

5. Section 508 Compliance
6. Reusable Technical Services
7. API Gateway Pattern
8. Open Standards and Open Source Based
9. NIEM 3.0 Compliance

SafeKids Prototype

A publicly accessible prototype is hosted at http://xfusionsafekids.org

GitHub Repository

https://github.com/xFusionTech/SafeKids (branch=master)

Technical Approach

This section describes how our technical approach aligns with various plays in the Digital Services Playbook, and provides supplementary listings of relevant documents provided in SafeKids GitHub. A consolidated list of artifacts for each requirement is provided in the deliverable on GitHub.

- *ADPQ-SafeKids-CriteriaEvidence.xlsx*

All documents mentioned here are located in the following folder in GitHub.

SafeKids/Final-Deliverables/

(a) We created a project charter, identifying one Project Lead who had responsibility for, and accountability of, the project. We decided that the Lead and Product Owner would be one person, responsible for the project scope, release planning, backlog priority, and required resources and funding.

Document(s):

- *ADPQ-SafeKids-Project Charter*
- *ADPQ-SafeKids-Scrum Team Organization*

(b) To allow the team to work independently and parallelly on multiple features, we took a micro-service based approach which ensured a modular approach for design, development and testing of application features independently in an iterative and incremental way.

We formed four groups that each had expertise to perform Front and Back End Web Development. Teams were responsible for one **Micro-service** which was developed and tested independently. Ultimately, the services were integrated with the frontend application.

Other Scrum Team Roles(Business Analyst, Technical Architect, Visual Designer and Usability Tester) are global to the individual service teams. For the Scrum Team, we designated a Scrum Master who doubledas the Agile Coach. The Product Owner, accountable for the quality of the product, was responsible for defining the product features and their priorities, sprint and release planning.

Our multidisciplinary team wascomprised of eleven Labor Categories.

Document(s):

- *ADPQ-SafeKids-Scrum Team Organization*
- *ADPQ-SafeKids-Project Charter*
- *ADPQ-SafeKids-Scrum Board And Burndown*

(c) Usability testing was conducted to ensure that the Graphical Design and User Experience met the users' expectations. It validated and confirmedour UX design principle – a human-centered design – with the goal of providing first-class experience using a responsive design.

A user (who impersonated a biological parent of a foster child), the Product Owner, Delivery Manager, Usability Tester, and GUX Designer were part of the testing (conducted in the conception and design phases) and UX Design process.Before being released to the development team, the GUX was updated based on the testing feedback.

Document(s):

- *ADPQ-SafeKids-User Personas*
- *ADPQ-SafeKids-Usability Testing*
- *ADPQ-SafeKids-Test Approach*
- *ADPQ-SafeKids-User Stories*

(d) We used six Human-Centered Design techniques:

- JAD Session and User Interviews
- Personas
- User Stories
- Screen Sketching
- Prototyping with Wireframes
- Usability Testing

Document(s):

- *ADPQ-SafeKids-User Personas*
- *ADPQ-SafeKids-Human Centered Design*
- *ADPQ-SafeKids-Usability Testing*
- *ADPQ-SafeKids-Test Approach*
- *ADPQ-SafeKids-User Stories*

(e) We useda Twitter Bootstrap design style that mirrored our vision for the prototype. As we matured our wireframes, we modified the styling to meet our UX needs. We developed our style libraries and design style guide, and provided the team with our specifications. We provided the developers wireframes and high-fidelity mockups to implement iteratively.

Document(s):

- *ADPQ-SafeKids-Style Guide*
- *ADPQ-Wireframes (multiple iterations)*
- *ADPQ - ADPQ-SafeKids-UI Prototype_Iteration (multiple iterations)*

(f) We conducted user interviews and usability tests with peoplefor persona types. We conducted several rounds of usability testing on the wireframes, subsequently modifying them based on testing feedback.

Document(s):

- *ADPQ-SafeKids-User Personas*
- *ADPQ-SafeKids-Usability Testing*
- *ADPQ-SafeKids-Section508Compliance*
- *ADPQ-Safekids-Section508ScanReport*
- *ADPQ-SafeKids-Test Approach*
- *ADPQ-SafeKids-User Stories*

(g) We developed SafeKids using an iterative and incremental approach where feedback informed subsequent versions of the prototype. In the Vision and Planning phase, we developed the UX/UI prototype that was tested for Usability Responsive Design iteratively, and features were improvised incrementally. We conducted three one-week sprints that iteratively developed four core product features in parallel. Described in our Test Approach, the product features were demonstrated in Sprint Review sessions, and garnered feedback to improve subsequent iterations.

Document(s):

- *ADPQ-SafeKids-Product Roadmap and Release Plan*
- *ADPQ-SafeKids Scrum Team Organization*
- *ADPQ-SafeKids-Scrum Board and Burndown*
- *ADPQ-SafeKids-Test Approach*
- *ADPQ-SafeKids-User Stories*

(h) We adopted responsive design tools and techniques to ensure the SafeKids application provided optimal user experience across multiple devices. We used Twitter Bootstrap framework for the frontend application, and AngularJS and JQuery to create a Single Page Application which provides easy reading, navigation and great user experience.At the beginning of the UX/UI design and Prototype Testing phases, the application was testedon multiple devices – from desktop computer to mobile devices, described in *ADPQ-SafeKids-Responsive Design*.

Document(s):

- *ADPQ-SafeKids-Test Approach*
- *ADPQ-SafeKids-Responsive Design*
- *ADPQ-SafeKids-Human Centered Design*

(i) We developed SafeKids using 13 modern, open-source technologies most appropriate to implement the prototype.

Document(s):

- *ADPQ-SafeKids-Architecture Document (Section 3.1)*

(j) We deployed SafeKids on the Amazon Web Service Elastic Compute Cloud (EC2) IaaS.

Document(s):

- *ADPQ-SafeKids-Architecture Document (Section 3)*

(k) We developed unit test cases for code, written with appropriate tooling for the technology stack. We used Open Source automated test tool, Selenium for automated Unit Testing/Regression Testing.

Document(s):

- *ADPQ-SafeKids-Test Approach*
- *ADPQ-SafeKids-Test Cases*
- *ADPQ-SafeKids-Automated Test Results*

(l) The Scrum Team worked parallelly on four application features in an iterative and incremental fashion. Each team's productwas checked-in to the source code control system (GitHub) twice/day – midday, and end of the day.

The built was performed after each check-in using Microsoft Visual Studio. Automated regression test was performed on the new built. If the test was successful, then the built was deployed in the DEV/TEST server where integration testing was performed. If the test failed, a notification was sent to the developers. The developer fixed the code, performed unit testing of the code, and checked-in the code to GitHub.Afterwards, a builtwas performed and regression test wasagain conducted. Integration Testing was performed in the TEST environment. After successful integration testing, the built was deployed in the PROD environment in the AWS environment.

Document(s):

- *ADPQ-SafeKids-Architecture Document (Section 5)*

(m) We utilized GitHub for distributed source control, and as document repository. We leveraged the GitFlow model of propagating code from feature branches, to development, release, and finally to production.

Document(s):

- *ADPQ-SafeKids-Architecture Document (Sections5-6)*

(n) Our development, maintenance, and operations included continually monitoring preventive and detective controls using industry standards. We usedthe Amazon Web Services environment for production, and used tools that it provides(AWSConfig, AWSCloudtrail) for continuous application monitoring.

Document(s):

- *ADPQ-SafeKids-Continuous Monitoring*

(o) We deployed SafeKids in Amazon Web Service Compute Cloud (AmazonEC2).

Document(s):

- *ADPQ-SafeKids-Architecture Document (Section 3)*

(p) Instructions to install and run our prototype are provided in the GitHub repository.

Document(s):

- *ADPQ-SafeKids-Installation.md*

(q) All code created for this project is licensed under the CC0 license. For further details and software used.

Document(s):

- *LICENSE.md*