As requested in RFI document, in this section we have provided a description of our approach adopted in creating the working prototype along screen workflow.

# **CHHS Prototype Development Approach**

AgreeYa as a CMMI certified organization brings over 17 years of experience of delivering complex and large projects leveraging Agile development methodology, templates and guidelines. We recognize each of the projects have a set of unique strategic objectives, business needs, operational considerations and organizational challenges. As a result, we believe in a collaborative, well-communicated and consultative approach

In this case as well the CHHS Prototype development was carried out following Agile Scrum methodology that provides ability to develop iteratively and incrementally. A multidisciplinary team led by experienced Product Owner/Project Manager carried out all the activities in order to successfully deliver a working CHHS prototype. The entire requirements elicitation, design and development process involved representatives from usergroups to work in collaboration with the scrum team.

During the prototype development Joint Application Development Sessions (JAD) were conducted to understand the need/requirement, deliberated functionality workflows, design screen mockups and documented Business Requirement Document (BRD). Basis this BRD, the product backlog -Feature/User stories, tasks were created with approved backlog priority in the Redmine tool. Daily standups, Task Boarding were done to review the progress of the work.

**Technical Approach and General Architectural considerations:**

* **Service Oriented Architecture** - The prototype is based on the Service oriented architecture (SOA) with the client-tier, business-tier and data-tier layers. The solution is built using AngularJS client-tier and Java backend with Jersey RESTful Web Services framework which is an open source, production quality, framework for developing RESTful Web Services in Java.
* **Loose Coupling** - All services are loosely coupled and autonomous to provide maximum flexibility in development, deployment and usage. Each service communicates with another through a service invocation and allow ease of communication without much concerns of service's internal implementation. Also loose coupling are implemented by layering the architecture in multiple tiers and segregating presentation, application and database layer.
* **Lightweight Data Exchange Format** - Data exchange between client-tier and business-tier (services) is through JSON objects. We believe JSON is a lightweight data interchange format and its use moves the processing of data (returned) on to the browser and hence reduces load on the server. Even though usage of JSON is recommended, we can alternatively just exchange plain XML data also.
* **Audit Trail** - At the backend layer the logging and tracing is handled at the controller level to capture the details of the entire API calls. The application log table will store these details.
* **Usability** - The user interfaces have dynamic structure that do not require complete page refreshes to complete tasks The CHHS application has ability to efficiently support data – minimizing i.e. user will have to follow minimum number of keystrokes to enter all required information. An AngularJs library is used for Ajax interactions at client layer and to build highly interactive rich UI web application.
* **Maintenance and Portability** – The whole application’s business logic and data can be packaged and transferred from one platform to other like local deployment to hosted solution, hosted solution to localized deployment, from one hosted solution to the other with minimum interruption. The application is developed using standard web technologies to provide full accessibility from desktop/laptop and tablet browsers.
* **Security –** As soon the user logs into the application a user specific session-Id is created which does not get time out until the subscriber gets inactive during configured duration. The handshake between client & API calls is enabled by sharing this session-Id in every subsequent API calls by that client. This secures client access in the application. Additionally, the application can be accessed over the http secure using ssl certificate for secure communication.
* **Sample Request Flow**

The client request coming from web browser, Android Tablet/Smartphone hits the Business-tier which is composed of java backend services (REST - WS). The java backend services would fetch the data and response back relevant data to the presentation layer. Data exchange between client and/or business services happens using JSON objects.

The CHHS prototype development follows standards of Java code conventions and checkstyle guides as prescribed by AgreeYa Solutions Design team. This enforces greater consistency in the code that makes it easier to develop and maintain. Integrating these guidelines with IDE facilitates clean and consistent code pattern that reduces build time issues.

# **Prototype Screen Workflows**

1. **Step 1 Registration -** User enters the URL , <https://fostercaredemo-agreeya.com> in the web browser or smartphone
2. **Step 2 Registration -** User lands on the Home page from where a new user can initiate activity “creation of a Family profile” by clicking on the Family Registration link.
3. **Step 3 Registration -** Family profile registration is a step by step process involves criteria’s to gauge Eligibility Criteria followed by Account Details, Personal Information, Family Narratives & Preferences, License & Agency Details tabs. On successfully completion of this task user can submit their Profile details which are saved into database. Once done, user receives an acknowledgement for completing the registration process with status message “Under review till approved”.

To facilitate user during registration process we have allowed user to partially fill the form, save and exit from the unfinished application and return anytime to continue filling the form with valid logins.

1. **Login -** An existing user ( with login credentials) can login and logout to the application by clicking the logion/LogOut link on the Home page
2. **Manage Profile -** An existing user can click on the Manage profiles link to update their approved profiles details anytime at later stages.
3. **Search & API Integration -** An existing user can search through the Children Residential Facilities within a given zip code area. Upon entering a zip code the application communicates with the HHS API to fetch data for all the facilities (Adoption Agency, Foster Family Agency and Foster Family Sub-Agency) within the entered zipcode and displays it in a grid on the screen.
4. **Locate Agency -** An existing user can search through the Foster Care Agencies in their nearby neighborhood. Upon entering a radius limit of users current geo location, the application communicates with the HHS API to fetch data for the facilities with selected Facility type of Foster Family Agency within the given coordinates range. The details of same are displayed in a grid on the screen.
5. **Private MailBox-** An existing user can view a Private mailbox.

The Prototype testing involved execution of automated Unit test cases using JUnits. Additionally, automation system test case suite is build to test the entire application. During sprint reviews the usergroup are involved to gather feedback around usability and accepted refinements are added as new backlog work items. User feedbacks are ultimate to successful release of the application.The build and deployment process is executed using the Jenkins Continuous integration tool. Successful builds(.war files) are deployed automatically to a target server. An automated smoke test case is executed to confirm the successful working of the application.