natomatechnologies
INCORPORATED

CALIFORNIA HEALTH & HUMAN SERVICES AGENCY

Architecture for Agile Development Pre-Qualified (ADPQ) Prototype - Draft

May 23, 2016

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Revision Record

Version Number	Status	Date	Sections	Change Description

Reviewers

Name	Position	Date

Approvals

Version Number	Status	Date

1. Introduction

1.1 Purpose

This Architecture document defines architectural approach and pattern implementation to be used in the ADPQ prototype.

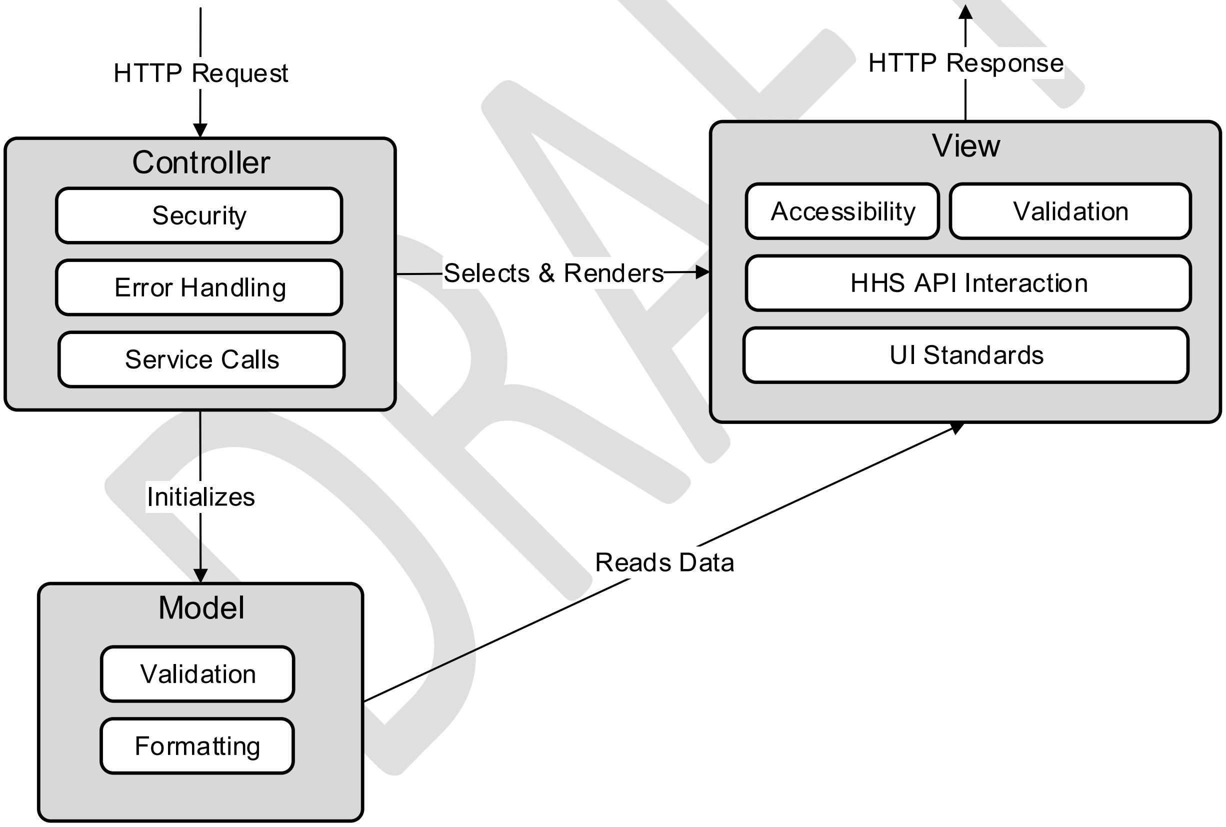
2. Patterns

This section describes the primary design patterns and other coding patterns used in the HHS prototype.

2.1 Model, View, Controller (MVC)

Use Microsoft's current implementation of an MVC framework, MVC 5. Figure 1 identifies which concerns are to be addressed at each component. Note that validation is performed within the view using jQuery on the browser to help provide a responsive user experience but also at the model within the web server to protect against users who might circumvent the UI validation rules. Additionally, the view performs all interactions with the HHS API using the Knockout libraries.

Figure 1 - Responsibilities within MVC



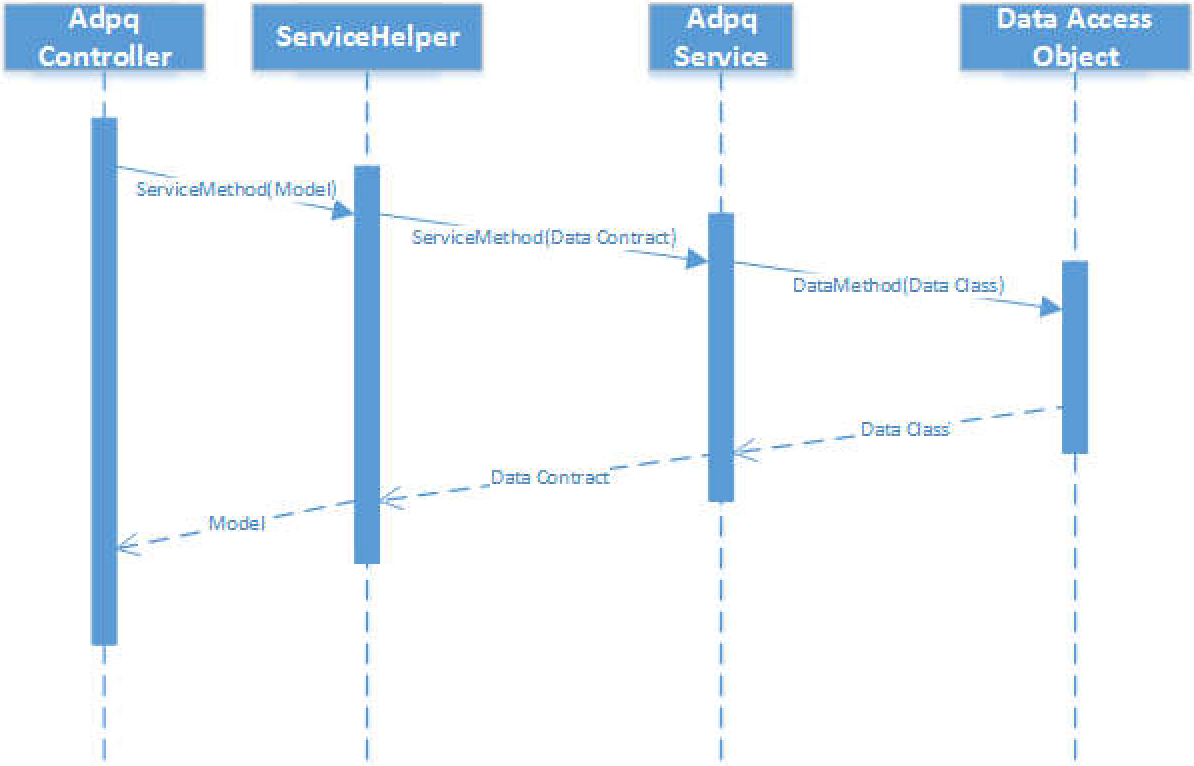
2.2 Dependency Injection (DI)

Follow the DI pattern to allow proper unit testing within the n-tiered architecture. Initially, implement both a constructor-based DI method and a default constructor that populates the dependencies. If time allows, integrate Unity and Unity.Mvc5 to provide the DI container functionality.

2.3 FAÇADE

In the MVC solution, use the ServiceHelper class for all interactions with the internal service. Implement the initial interface with the internal service using Microsoft's web reference technology. If time allows, configure the necessary components for Unity to discover the web service. ServiceHelper is responsible for translating between UI Model classes and service data contracts and for making the data service calls.

Figure 2 - Data flow between layers using the ServiceHelper façade



3. Architectural Layers

This section identifies the key responsibilities of each architectural layer and the tools that should be used to support those responsibilities.

3.1 Browser

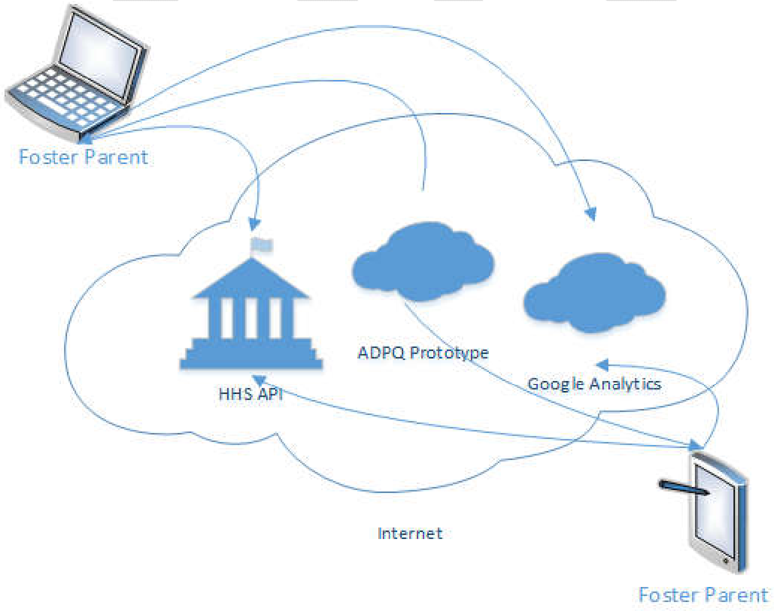
Table 1 lists the tools to be used within the client browser and what each tool is used for.

Table 1 - Browser Toolset

Tool	Use
Knockout	Interact with HHS API
Bootstrap	Framework for reactive layout
jQuery	Framework for browser based validation and other UI features
HTML5	Markup language for structuring and presenting content (initially generated by Razor scripts)
CSS3	Provide consistent look and feel
WAI-ARIA tags	Provide additional support to assistive technologies
HHS API	Provide details about children's residential facilities within a zip code
Google Analytics	Provide details about website usage for human interaction analysis

As shown in Figure 3, the foster parents receive web pages from the ADPQ prototype and then code on those web pages makes calls to the HHS API and Google Analytics directly from the user's browser.

Figure 3 - Interactions with HHS API and Google Analytics



3.2 User Interface (UI) Layer

Table 2 lists the tools to be used within the UI layer and what each tool is used for. For the purpose of this document, this covers all of the MVC implementation

Table 2 – User Interface Toolset

Tool	Use
Razor	.NET language for layout definition and UI logic
AutoMapper	Translate between UI Models and data transport classes
C#	.NET language for Controllers, Models, and logic within Views
Resource files	Culture aware .NET constructs for support of internationalization, if time allows

3.3 Service Layer

Table 3 lists the tools to be used within the UI layer and what each tool is used for.

Table 3 – Service Layer Toolset

Tool	Use
C#	.NET language
WCF	.NET technology for providing web services

3.4 Data Layer

Table 4 lists the tools to be used within the UI layer and what each tool is used for.

Table 4 – Data Layer Toolset

Tool	Use
C#	.NET language

Note: For this prototype, the data layer is implemented using simple serialization to XML files. The architecture obscures the implementation of the data layer from all other layers so migrating to a SQL database, noSQL database, or other persistence construct will not affect any other layer.

Appendix A - Americans with Disabilities Act (ADA)

Compliance with the ADA is performed by following the guidelines identified here.

SECTION 508 GUIDELINES

Table 5 – Section 508 describes the general approach to satisfying Section 508 requirements.

Table 5 – Section 508

Section 508 Standards Guide, Subpart B (Technical Standards) § 1194.22 Web-based intranet and internet information and applications	ADPQ Prototype Compliance Guidelines
(a) A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content).	Use an equivalent text element or tag which screen readers can read for each non-text element in a VoteCal developed web site page. Use an empty alt attribute (i.e. alt="") for purely decorative elements. Note: Do not use longdesc as none of the major browsers support it.
(b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.	Do not include any multimedia presentations.
(c) Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.	For web pages that show information conveyed with color, also convey this information a non-color equivalent. For example, if presenting something important in a red color, also have the same "important" description available as text or markup. So when displaying an Alert marked in red text also include the word "Alert" or tag with <strong> or <em>.
(d) Documents shall be organized so they are readable without requiring an associated style sheet.	Organize documents such that they are readable without requiring an associated style sheet. Design and implemented each web page to be readable with the style sheet turned off.
(e) Redundant text links shall be provided for each active region of a server-side image map.	Do not include any image maps.
(f) Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.	Do not include any image maps.
(g) Row and column headers shall be identified for data tables.	In data tables, use row and column headers (<th> tags) were applicable.
(h) Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.	For simple tables, use <th> tags marked with a scope of "row" or "col" as appropriate. For complex tables, use <th> tags marked with an ID then identify the associated <td> cells as belonging to its <th> by using the header attribute.

Section 508 Standards Guide, Subpart B (Technical Standards) § 1194.22 Web-based intranet and internet information and applications	ADPQ Prototype Compliance Guidelines
(i) Frames shall be titled with text that facilitates frame identification and navigation.	Do not use frames.
(j) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.	Do not include any elements on the web pages that flicker as a part of the web design. For example: do not include any animated gif images that flicker.
(k) A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.	When unable to comply with these guidelines, create a text-only page with equivalent information or functionality.
(l) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.	Where using scripting to display content or to display interface elements, tag this content such that that assistive technology can read it. When this is not possible, identify the information provided by the script with functional text that assistive technology can read.
(m) When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with §1194.21(a) through (l).	Do not use applets or plug-ins for internal or public web pages.
(n) When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.	Where using a form or page to collect information such as for the user profile, allow people using assistive technology to access the information, field elements, and functionality as required.
(o) A method shall be provided that permits users to skip repetitive navigation links.	Include a link to main content.
(p) When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.	Do not require timed responses from web users. Provide a session timeout warning that allows users to reset the session timeout clock.

WCAG 2.0 GUIDELINES

Table 6 – WCAG 2.0 Guidelines describes the general approach to satisfying the WCAG 2.0 guidelines.

Table 6 – WCAG 2.0 Guidelines

WCAG 2.0 GUIDELINE	VOTECAL COMPLIANCE GUIDELINES
Perceivable – Provide text alternatives for non-text content. Provide captions and other alternatives for multimedia. Create content that can be presented in different ways, including by assistive technologies, without losing meaning. Make it easier for users to see and hear content.	See guidelines for (a), (b), (c), (e), (f), and (k) in Table 5 – Section 508.
Operable - Make all functionality available from a keyboard. Give users enough time to read and use content. Do not use content that causes seizures. Help users navigate and find content.	Only use keyboard accessible controls. See guidelines for (p), (j), and (0) in Table 5 – Section 508.
Understandable- Make text readable and understandable. Make content appear and operate in predictable ways. Help users avoid and correct mistakes.	Design pages to keep content succinct and understandable. Design each web page to contain only enough information for the user to understand what is needed and how to provide the information.
Robust- Maximize compatibility with current and future user tools.	Build the UI using HTML 5 and CSS3.