System Architecture

# Introduction

The purpose of this document is to briefly explain the architecture of the Age Calculation System. This system has been designed as two main components, the Server Side Solution and the Front End Solution.

# Server Side Solution

The purpose of the server-side solution is to facilitate the storage of the user’s data on a cloud environment (also known as online storage). In order to achieve this goal, I have created a Database using Microsoft’s SQL Server 2008 (release 2). This database is stored on an online server running on a Microsoft Windows platform. I have also created a program called “AgeCalc”. “AgeCalc” has been coded using Microsoft’s C# programming language. This program runs on the online server listening for data requests from other platforms (Websites, mobile devices etc.). When a platform e.g. a website, requests for data, this program breaks the data into C# objects and then processes them accordingly. The program makes use of the Entity Framework for accessing the database.

The communication between the platforms and the server is achieved by using a Hypertext Transfer Protocol (HTTP). However, Secured Socket Layer (SSL) can very easily be switched on, as the server already implements an SSL Certificate. All that needs to change is that the HTTP is restricted and the platforms use HTTPS instead when requesting the data. This means that the data transferred/communicated will always be encrypted.

## Important Components

The most important components in this solution are

* User Age Controller (UserAgeController.cs)
* Hash Date (HashDate.cs & HashDateDefinitions.cs).

Together these components are responsible for calculating the user’s age and storing it.

# Front End Solution

The front-end solution is the user graphical interface (GUI) through which the user interacts with the system. This solution allows the user to enter their details and retrieve their calculated age from the server. The front end is also responsible for validating the data entered. It makes sure only the data, which has passed validation checks is sent to the server in order to avoid any issues.

The front end has been designed to function independently from the server side reading and writing of the data. The Web API (web services) created on the server are called upon by the front end in order to calculate the user’s age and add them to the list of previous visitors.

The data is fetched asynchronously from the server, thereby allowing the web page to remain interactive.

The front end has been programmed as an HTML page, which makes use of the following technologies:

## CSS 3 Styling

CC3 styling has been used in order to facilitate the animations of the pen icon, sending the editing controls flying across the screen, and for the submit button. All colours and other styling are controlled via CSS.

## jQuery

The jQuery library has been used for easy manipulation of JavaScript. The Angular JS framework also depends upon this library.

## Angular JS

I have used Google’s Angular JS framework for giving the complicated JavaScript code, a C language like structure. The advantage is Angular JS is that it provides box-of-the-box functionality for performing validation checks of data entered by the user. Angular JS binds HTML pages, CSS and JavaScript files into a Model-View-Controller (MVC) architecture. This allows developers to write easy to manage and modular code.

## BootStrap

I have used Bootstrap for styling the input controls and their validation messages. Bootstrap is also vital for the functioning of the Date picker control. Without Bootstrap, this control will not work.

## SoapClient

The Soap client library allows the front end to make Web API data request calls to the server. It parses the data received into JavaScript objects. The Soap client library was not originally designed to fetch data in an asynchronous manner. I had to modify this library in order to achieve asynchronous fetching of data.

A lot of custom CSS styling and JavaScript has been used in order to make the page both graceful and interactive.

## Supported Browsers

While all recent browsers are supported, it is highly recommended that you view the page using the latest Chrome or Firefox browser. There are some CSS effects, which do not work well on Internet Explorer.