# SOFTWARE DESIGN DOCUMENT (SDD)

## ANTHROCLOUD

Software Modernization for Cloud-based Pediatric Anthropometry

December 4, 2019

Version 2

COT 6931

Dusty Wright

# Change History

| Version | Date | Author | Changes |
|---|---|---|---|
| 0.1 | July 21, 2019 | Dusty Wright | Initial Document |
| 1.0 | July 31, 2019 | Dusty Wright | Baseline |
| 2.0 | December 4, 2019 | Dusty Wright | 1.1 Clarify software design<br>1.5 Clarify section 4 and 5 purpose<br>2 module ownership<br>3.4 Added “correlative”<br>3.4.2.1 Added script filenames<br>3.8 Added MVC & Google Charts<br>3.8 Table 1 updated<br>4 Removed unused tables<br>5.1 Data Entry - Figure 7 entry form updated<br>5.2 Input Elements - Figure 8 User Interface Validation updated |

# Table of Contents


1 Introduction ..... 4
1.1 Purpose ..... 4
1.2 Scope ..... 4
1.3 Definitions, acronyms, abbreviations ..... 4
1.4 References ..... 4
1.5 Overview ..... 5
2 Design Overview ..... 5
3 Architectural Design ..... 5
3.1 User View ..... 5
3.1.1 Get BMI ..... 5
3.1.2 Get Age ..... 5
3.1.3 Get Z-Score ..... 5
3.1.4 Get Percentile ..... 5
3.1.5 View Chart ..... 6
3.1.6 Assess Status ..... 6
3.2 Process View ..... 6
3.2.1 Get BMI ..... 6
3.2.2 View Chart ..... 6
3.3 Physical View ..... 7
3.4 Development View ..... 8
3.4.1 Assembly ..... 8
3.4.2 Namespaces ..... 8
3.4.3 Folder ..... 9
3.5 Logical View ..... 10
3.5.1 Components ..... 10
3.5.2 Interfaces ..... 11
3.6 Size and Performance ..... 12
3.7 Quality ..... 12
3.8 Design Rationale ..... 13
4 Data Design ..... 13
4.1 Data Description ..... 14
4.1.1 Length/height-for-age ..... 14

4.1.2 Weight-for-age 14
4.1.3 Weight-for-length 14
4.1.4 Weight-for-height 14
4.1.5 BMI-for-age 14
4.1.6 Head circumference-for-age 15
4.1.7 Arm circumference-for-age 15
4.1.8 Subscapular skinfold-for-age 15
4.1.9 Triceps skinfold-for-age 15
5 Human-Computer Interface Design 17
5.1 Data Entry 17
5.2 Input Elements 17
5.3 Chart Elements 18
5.4 Color Coding 19
6 Requirements Traceability 20

# 1 Introduction

## 1.1 Purpose

The purpose of this document is to specify architecture and design for the AnthroCloud software solution. The intended audience for this document is both stakeholders and designers. This document captures and constructs software design from requirements.

## 1.2 Scope

The AnthroCloud software solution is a modern cloud-based WHO Child Growth Standards compliant anthropometric calculator designed to provide a more flexible, maintainable, and portable solution to meet the changing needs of users. The scope of this project is limited to WHO Child Growth Standards, z-score calculations, percentile calculations, growth curves, and plotted scores.

The objectives of the projects are as follows:

1. Choose an application platform to improve the ability of the calculator to change to new specifications or operating environments.
2. Design software that decouples the application into components to reduce the effort necessary to change the calculator.
3. Provide a common interface to better exchange data between calculator components.
4. Build a test suite to reduce the effort necessary to verify calculator changes.

## 1.3 Definitions, acronyms, abbreviations

| | | |
|---|---|---|
| API | Application Programming Interface | A publicly available web-based application programming interface that accesses functions to returns data. |
| App | Application | An application is referred to as "app" in this context is a type of application software designed to run in the Cloud. |
| BMI | Body Mass Index | Body mass index (BMI) is a height and weight body fat measure. |
| HTML | HyperText Markup Language | A markup standard that defines properties and behaviors of web page content. |
| HTTP | HyperText Transfer Protocol | HTTP is a protocol for transferring data over the Web. |
| IDE | Integrated Development Environment | An application that provides software development resources. |
| JAR | Java Archive | A package file format that is an aggregate of many java class files. |
| MVC | Model-View-Controller | A design pattern for decoupling the user-interface, data, and controller. |
| REST | Representational State Transfer | A set of architectural constraints for web services that define standards for communication. |
| SQL | Structured Query Language | A language standard for communicating with databases. |
| UI | User Interface | The screens and visual elements that interface with the software. |

## 1.4 References

Kruchten, P. (1995). Architectural Blueprints - The "4+1" View Model of Software Architecture. *IEEE Software 12*(6), 42-50.

Otero, C. E. (2012). Software Engineering Design Theory and Practice. Boca Raton, FL: CRC Press.

### 1.5 Overview

The next sections lay out a hierarchical solution design. Section two gives a design overview. In section three, a high-level view of software architecture is presented using views of the architectural model. Detailed designs are provided to compliment the model. Sections four and five provide additional detailed design to meet requirements.

# 2 Design Overview

The WHO ANTHRO desktop application was published in 2006 by the World Health Organization (WHO). Its Anthropometric Calculator module represents the core functions necessary for calculation. The module calculates pediatric anthropometrics based on WHO Child Growth Standards. The results are used to determine nutritional adequacy of infants and young children. The executable application is free for download; however, it is not open source and no application programming interface is available. The software does not provide a web application, web service, or test suite.

The WHO ANTHRO desktop application presents the same issues faced by most WIC enterprise software efforts. While not as concrete as the 1990 EpiInfo dBase CDC Anthro project, the WHO ANTHRO software technology is old and not easy to move. How can better software design and newer technology reduce the repeated effort to calculate pediatric anthropometrics? This project will create a modern cloud-based WHO child growth standard-compliant anthropometric calculator designed to provide a more flexible, maintainable, and portable solution to meet the changing needs of users.

# 3 Architectural Design

Kruchten's *4+1 Architectural View Model* addresses the issue of "prematurely partitioning the software," by concentrating software architecture on concurrent views (Kruchten, 1995). The architectural design includes five main views: logical view, process view, development view, physical view, and user view.

## 3.1 User View

The user view of the software is represented by a use case diagram in Figure 1. The system is the AnthroCloud software solution. The growth monitor represents the user operating the calculator to monitor child growth and nutritional status. The other actors represent a subset of system responsibilities. The major actions that can be taken by the user are listed below.

### 3.1.1 Get BMI

The user enters height and weight and expects a calculated BMI in return.

### 3.1.2 Get Age

The user enters date of birth and date of visit and expects a calculated age to return. The returned value is formatted for display in year, months, and total months. The age in days value is necessary for internal calculation of anthropometric indices.

### 3.1.3 Get Z-Score

The user completes data entry and views the calculated z-score for eight anthropometric indices.

### 3.1.4 Get Percentile

The user completes data entry and views the calculated percentile for eight anthropometric indices.

#### 3.1.5 View Chart
The user pulls up a chart for eight anthropometric indices to view the graphic display of the measurement.

#### 3.1.6 Assess Status
The user completes data entry and views a color-coded calculated z-score or calculated percentile for eight anthropometric indices. The color coding instantly informs the user of nutritional status.

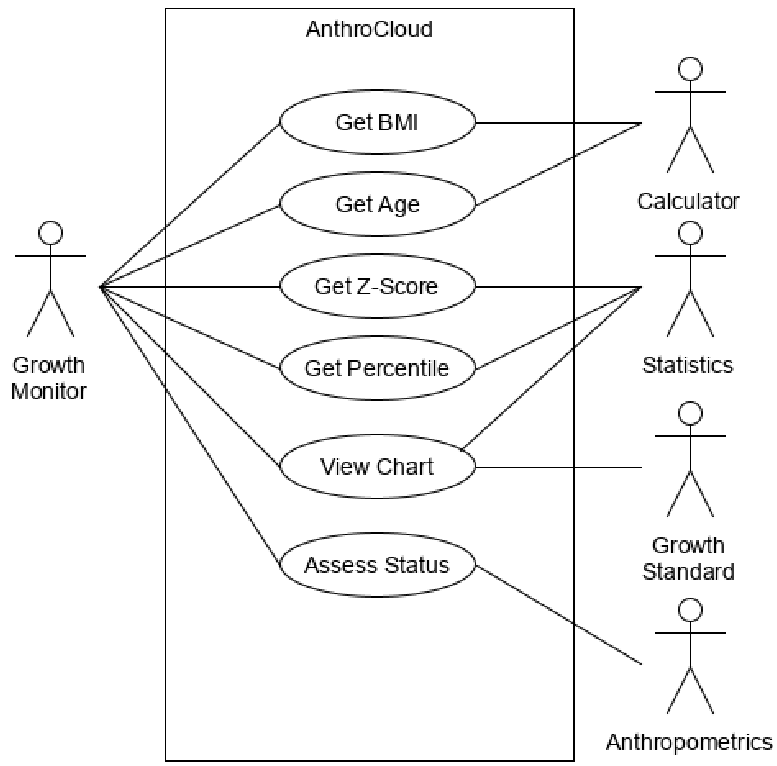


*Figure 1 Use Case Diagram*

### 3.2 Process View
The process view of the software is represented by a sequence diagram in Figure 2. The user enters BMI data and opens a chart. The sequence processes and waits for information. BMI is calculated after data is entered. A BMI Chart is viewed. The user enters height and weight and then opens a chart.

#### 3.2.1 Get BMI
The Get BMI use case scenario activities are expanded in the Process View sequence diagram above. Height and weight are necessary to calculate a BMI value. Once entered, the BMI is calculated and returned. The BMI Z-Score and Percentile are also calculated, returned, and displayed during this time.

#### 3.2.2 View Chart
The View Chart use case scenario activities are expanded in the Process View sequence diagram above. The z-score and percentile are used to compare against retrieved BMI-for-age growth data for similar

age and sex. The data is retrieved and displayed. The retrieved data is used to draw growth curves. The score and percentile are used to plot a point on the chart.

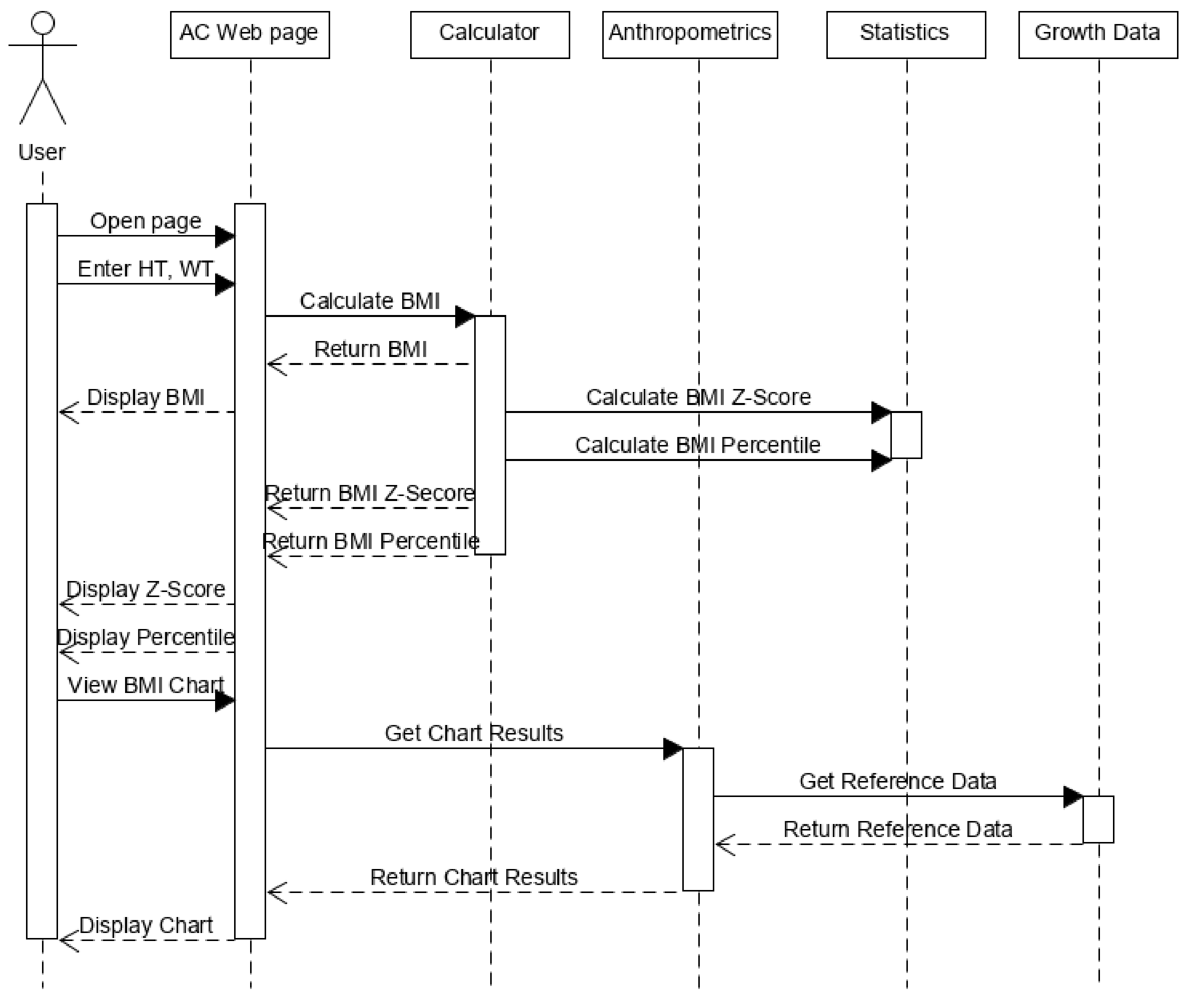


*Figure 2 Sequence Diagram*

## 3.3 Physical View

The physical view of the software is represented by a deployment diagram in Figure 3. The deployment device targeted by the software solution is the Azure Resource Group. This grouping represents resources available to the Azure subscription plan. The Azure App Service deployment can run multiple application types within a single instance. These nodes are depicted separately to show different application types within the Azure App Service. The applications run on managed Virtual Machines. The web application is represented as an artifact and deployed to the Azure Application Service as a Web App. The web service is represented as an artifact and deployed to the Azure Application Service as an API App. The database artifact is deployed to the Azure SQL Database. The web application is rendered to the client's web browser over HTTP. Requests pass through the web application, then web service and finally the database. The deployment diagram identifies the elements comprising the physical view of the system. The solution shall contain a Dockerfile to support configurable deployment of an image to Azure.

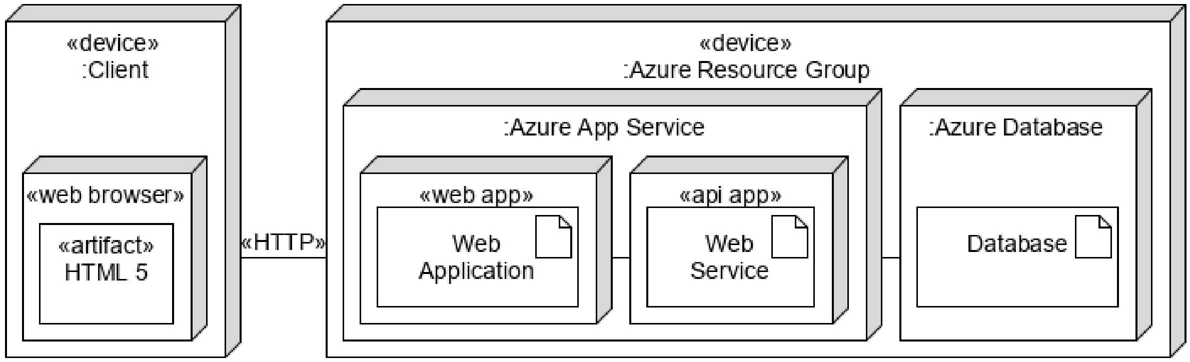


*Figure 3 Deployment Diagram*

## 3.4 Development View

The development view is represented as a deployment diagram in Figure 4. This view is used to show how logical components are physically organized. Namespace and Package are correlative concepts in the .NET Framework as well as Assembly and Java's executable Java Archive File (JAR). The physical files and directories are dictated by the following namespaces comprising the web application and web service assemblies. Additional content placed in a folder is documented.

### 3.4.1 Assembly

The diagram uses an UML frame to capture the grouping of related components within the assembly.

#### *3.4.1.1 Web Application*

The web application assembly is comprised of the AnthroCloud.UI namespace.

#### *3.4.1.2 Web Service*

The web service assembly is comprised of the AnthroCloud.Service, AnthroCloud.Business, and AnthroCloud.Data namespaces.

### 3.4.2 Namespaces

Namespaces are unique and organize code into logic groups. All namespaces begin with the solution name AnthroCloud.

#### *3.4.2.1 AnthroCloud.UI*

All user interface elements are found in the AnthroCloud.UI namespace. The AnthroCloud.UI.Web namespace will contain the single data entry form for the project. The AnthroCloud.UI.Chart namespace will contain the chart used to draw the eight anthropometric indices. AnthroCloud.UI will consume AnthroCloud.Service. This namespace is effectively the presentation layer.

#### *3.4.2.2 AnthroCloud.Service*

The AnthroCloud.Service contains the interface and HTTP namespaces. AnthroCloud.Service.HTTP represents the services and methods consumed by the user interface. AnthroCloud.Service.Contracts represents the contracts necessary to consume the interface. This AnthroCloud.Service namespace is effectively the service layer. AnthroCloud.Service will consume AnthroCloud.Business.

#### 3.4.2.3 AnthroCloud.Business

The AnthroCloud.Business contains the Calculator, Anthropometrics, and Statistics namespaces. The AnthroCloud.Business.Calculator contains all business calculator functions including anthropometrics functions and statistics functions. The AnthroCloud.Business.Anthropometrics namespace will contain business logic related to anthropometric results and analysis. The AnthroCloud.Business.Statistics namespace will contain business logic related to statistical calculation. The AnthroCloud.Business namespace is effectively the business layer. AnthroCloud.Business will consume AnthroCloud.Data.

#### 3.4.2.4 AnthroCloud.Data

The AnthroCloud.Data namespace is effectively the data layer. This namespace will communicate with the database. This is where growth reference data is retrieved.

#### 3.4.2.5 AnthroCloud.Test

The AnthroCloud.Test namespace is effectively the cross-cutting layer. Test would be depicted as a sidecar layer with the ability to access all layers vertically in a layered architecture diagram.

#### 3.4.2.6 AnthroCloud.Database

Data source refers a cloud-based SQL Database. An AnthroCloud.Database namespace could be created to house any related files outside of creation scripts.

#### 3.4.2.7 AnthroCloud.Entities

AnthroCloud.Entities is a shared namespace. Shared namespaces are not depicted in the diagram but described as a designed container for shared business logic. AnthroCloud.Entities could be used to house common business models accessible to each layer. This would be represented as a shared layer sidecar in a layered architecture diagram.

#### 3.4.2.8 AnthroCloud.Framework

AnthroCloud.Framework is a shared namespace. Shared namespaces are not depicted in the diagram but described as a designed container for cross-cutting concerns. AnthroCloud.Framework could be used to house common cross-cutting concerns accessible to each layer. This would be represented as a cross-cutting sidecar in a layered architecture diagram.

### 3.4.3 Folder

A folder can be used to house content for the solution. An example would be the Scripts folder.

#### 3.4.3.1 Scripts

A scripts folder will be created to contain the script for “AnthroCloudDB_Tables.sql” and “AnthroCloudDB_Tables_Expanded.sql”.

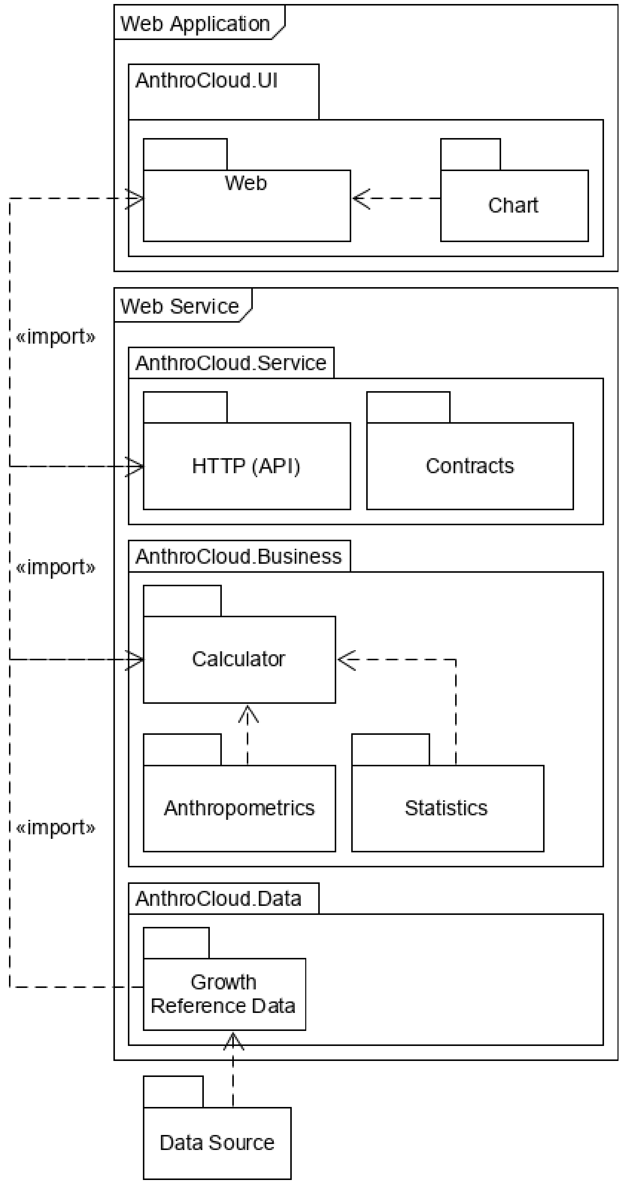


*Figure 4 Deployment Diagram*

## 3.5 Logical View

The logical view is represented as a component structure diagram in Figure 5. The AnthroCloud software is decomposed into logical components. The web application, web service, and database represent the highest component level. This view shows where components reside within the solution and the tasks performed.

### 3.5.1 Components

The logical system components are detailed below and separated from interfaces in the next section.

#### *3.5.1.1 Calculator User Interface*

This is the single web page for data entry.

##### 3.5.1.2 Chart User Interface Component
This is the chart control that will display the growth data curves and plots.

##### 3.5.1.3 Calculator Application Programming Interface
This is the AnthroCloud HTTP service. Data can only be access through the Web API.

##### 3.5.1.4 Calculator Business Object
This is the modular business container for all calculator business logic.

##### 3.5.1.5 Anthropometric
This component will contain business logic related to anthropometric results and analysis. Age and BMI calculations will be performed here.

##### 3.5.1.6 Statistics
This component will contain business logic related to anthropometric statistical calculation. Z-score and Percentile calculations will be performed here.

##### 3.5.1.7 Calculator Data Access
This is the modular data access container for all calculator data access logic.

##### 3.5.1.8 Growth
This is the data access object for retrieving standard growth reference data.

##### 3.5.1.9 Database
This is where the WHO Child Growth Standard data tables are stored.

### 3.5.2 Interfaces
The Figure 4 Component Structure Diagram depicts interfaces. The interfaces are described below.

##### 3.5.2.1 iChart
An interface is necessary to abstract away any functional dependence on a single chart assembly, third-party, or library.

##### 3.5.2.2 iService
An interface is created to describe all the operations available in the system.

##### 3.5.2.3 iCalculator
A business interface is created to describe all Calculator operations available in the system. The interfaces are depicted using a ball-and-socket notation.

##### 3.5.2.4 iData
A data interface is created to describe all Data operations available in the system.

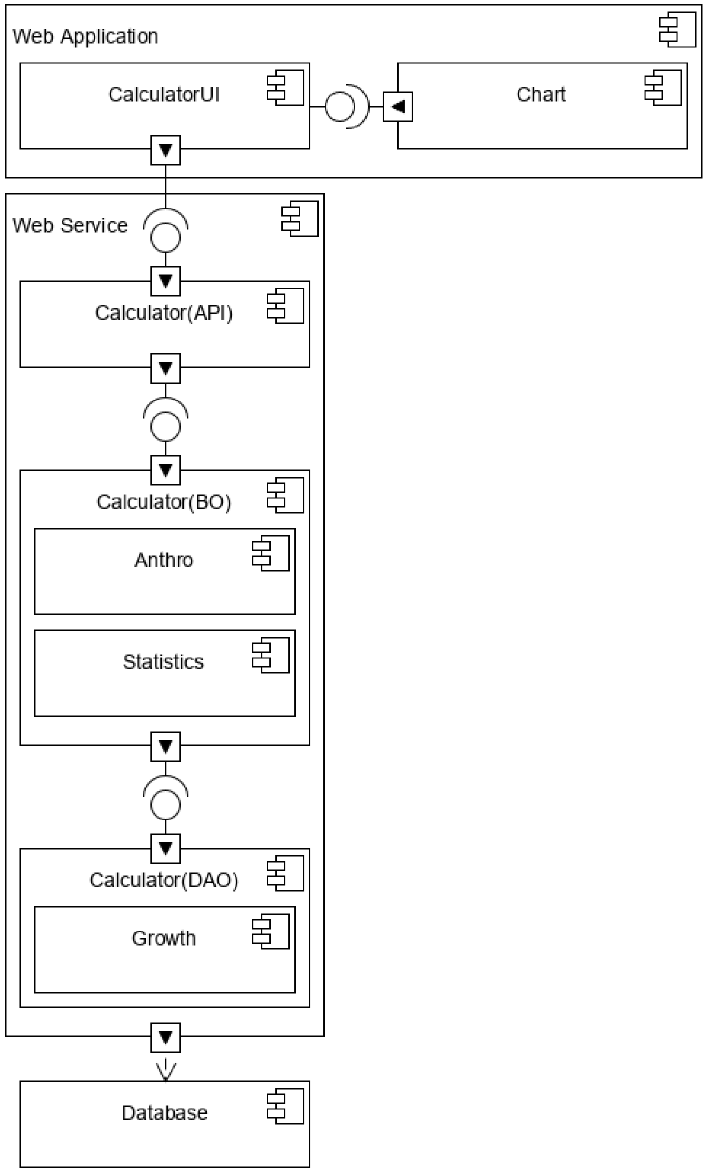


*Figure 5 Component Structure Diagram*

## 3.6 Size and Performance

The software architecture supports compliance with the stipulated performance requirements.

1. PER-1: The service shall not exceed 5 seconds for the longest running call.

## 3.7 Quality

The software architecture supports compliance with quality standards.

2. MNT-1: The solution shall use an application, service, and database.
3. MNT-2: The solution shall be architected using layers.
4. PRT-1: The solution shall use the .NET Core framework.
5. PRT-2: The solution shall use a container.

## 3.8 Design Rationale

A layered architectural pattern was selected for the design. The desirable quality attributes associated with this pattern are as follows: modifiability, portability, security, reusability (Otero, 2012, Pg. 135). The AnthroCloud uses multiple layers that separate system components. Layers contain functional components. The design decreases coupling and increases maintainability. Reuse becomes easier as layers and their components are compartmentalized. From presentation layer to data layer, communication between layers can be depicted using a single arrow indicating a *uses* statement or one-way relationship. This pattern is expected to offer portability necessary to deploy a web application and web service.

An interactive system pattern or Model-View-Controller was considered as a design alternative. The desirable quality attributes associated with this pattern are as follows: modifiability, usability, and reusability (Otero, 2012, Pg. 130). Views would reside in the presentation layer, models in the business, and controller, if properly abstracted, would also reside in the presentation layer. MVC provides the user with a simple and consistent view of the front-end. The user would not be aware of changes made to the backend as updates and upgrades are released. MVC is a strong second choice and does not preclude the selection and use of both patterns.

Preferred technology in Table 1 is described by layer and may be used for the AnthroCloud software solution. Visual Studio 2019 is chosen for its best-in-class integrated development environment. The IDE is used to create and manage the GitHub repository. The development platform for the cloud is .NET Core. Blazor is favored for this project as a preferred technology for building a client-side Web User Interface. Blazor can run client-side C# directly in the browser using WebAssembly. Automatic calculation of inputs into the calculator using real-time web functionality is desirable. ASP.NET Web API is chosen for adherence to REST API architecture and interoperability. Entity Framework Core is chosen for data access. Azure SQL Database is the database provider. HTML 5.0 must be used. Google Charts is an example of a web library that has a line chart with plot lines.

*Table 1 Preferred Technology*

| **Layer** | **Technologies** |
|---|---|
| Presentation | .NET Core, ASP.NET Core MVC, Google Charts, HTML5, Javascript |
| Service | ASP.NET Core Web API |
| Business | C# |
| Data | Entity Framework Core |
| Database | Azure SQL Database |

# 4 Data Design

The WHO Child Growth Standards tables are modeled as a class diagram in Figure 6. The growth reference data tables are defined by sex, z-score, percentile, various age indicators, and anthropometrics indicators. Sixty tables total are represented. For example, to accurately depict a nine-week old girl's length-for-age measurement, growth data for girls z-score from birth to 13 weeks should be used. Figure 6 shows an approach to consolidate this design into fewer physical containers. The tables are not transactional and have no relation. The data are static. The data are to be referenced.

## 4.1 Data Description

### 4.1.1 Length/height-for-age

The following data tables are stored in LHFA table:

1. Girls z-scores Length-for-age: Birth to 2 years
2. Girls z-scores Height-for-age: 2 to 5 years
3. Boys z-scores Length-for-age: Birth to 2 years
4. Boys z-scores Height-for-age: 2 to 5 years
5. Girls percentiles Length-for-age: Birth to 2 years
6. Girls percentiles Height-for-age: 2 to 5 years
7. Boys percentiles Length-for-age: Birth to 2 years
8. Boys percentiles Height-for-age: 2 to 5 years

### 4.1.2 Weight-for-age

The following data tables are stored in the WFA table:

1. Girls z-scores Weight-for-age: Birth to 5 years
2. Boys z-scores Weight-for-age: Birth to 5 years
3. Girls percentiles Weight-for-age: Birth to 5 years
4. Boys percentiles Weight-for-age: Birth to 5 years

### 4.1.3 Weight-for-length

The zero to twenty-four months data is given its own table for the length column. The following data tables are stored in the WFL table:

1. Girls z-scores Weight-for-length: Birth to 2 years
2. Boys z-scores Weight-for-length: Birth to 2 years
3. Girls percentiles Weight-for-length: Birth to 2 years
4. Boys percentiles Weight-for-length: Birth to 2 years

### 4.1.4 Weight-for-height

The zero to twenty-four months data is given its own table for the height column. Such a physical design separation makes it easier to support CDC tables from 24 months and up. This effort is just concerned with WHO, but the design choice is explained. The following data tables are stored in this table:

1. Girls z-scores Weight-for-height: 2 to 5 years
2. Boys z-scores Weight-for-height: 2 to 5 years
3. Girls percentiles Weight-for-height: 2 to 5 years
4. Boys percentiles Weight-for-height: 2 to 5 years

### 4.1.5 BMI-for-age

The following data tables are stored in the BFA table:

1. Girls z-scores BMI-for-age: Birth to 2 years
2. Girls z-scores BMI-for-age: 2 to 5 years
3. Boys z-scores BMI-for-age: Birth to 2 years

4. Boys z-scores BMI-for-age: 2 to 5 years
5. Girls percentiles BMI-for-age: Birth to 2 years
6. Girls percentiles BMI-for-age: 2 to 5 years
7. Boys percentiles BMI-for-age: Birth to 2 years
8. Boys percentiles BMI-for-age: 2 to 5 years

#### 4.1.6 Head circumference-for-age

The following data tables are stored in the HCFA table:

1. Girls z-scores Head circumference-for-age: Birth to 5 years
2. Boys z-scores Head circumference-for-age: Birth to 5 years
3. Girls percentiles Head circumference-for-age: Birth to 5 years
4. Boys percentiles Head circumference-for-age: Birth to 5 years

#### 4.1.7 Arm circumference-for-age

The following data tables are stored in the ACFA table:

1. Girls z-scores Arm circumference-for-age: 3 months to 5 years
2. Boys z-scores Arm circumference-for-age: 3 months to 5 years
3. Girls percentiles Arm circumference-for-age: 3 months to 5 years
4. Boys percentiles Arm circumference-for-age: 3 months to 5 years

#### 4.1.8 Subscapular skinfold-for-age

The following data tables are stored in SSFA table:

1. Girls z-scores Subscapular circumference-for-age: 3 months to 5 years
2. Boys z-scores Subscapular circumference-for-age: 3 months to 5 years
3. Girls percentiles Subscapular circumference-for-age: 3 months to 5 years
4. Boys percentiles Subscapular circumference-for-age: 3 months to 5 years

#### 4.1.9 Triceps skinfold-for-age

The following data tables are stored in the TSFA table:

1. Girls z-scores Triceps circumference-for-age: 3 months to 5 years
2. Boys z-scores Triceps circumference-for-age: 3 months to 5 years
3. Girls percentiles Triceps circumference-for-age: 3 months to 5 years
4. Boys percentiles Triceps circumference-for-age: 3 months to 5 years

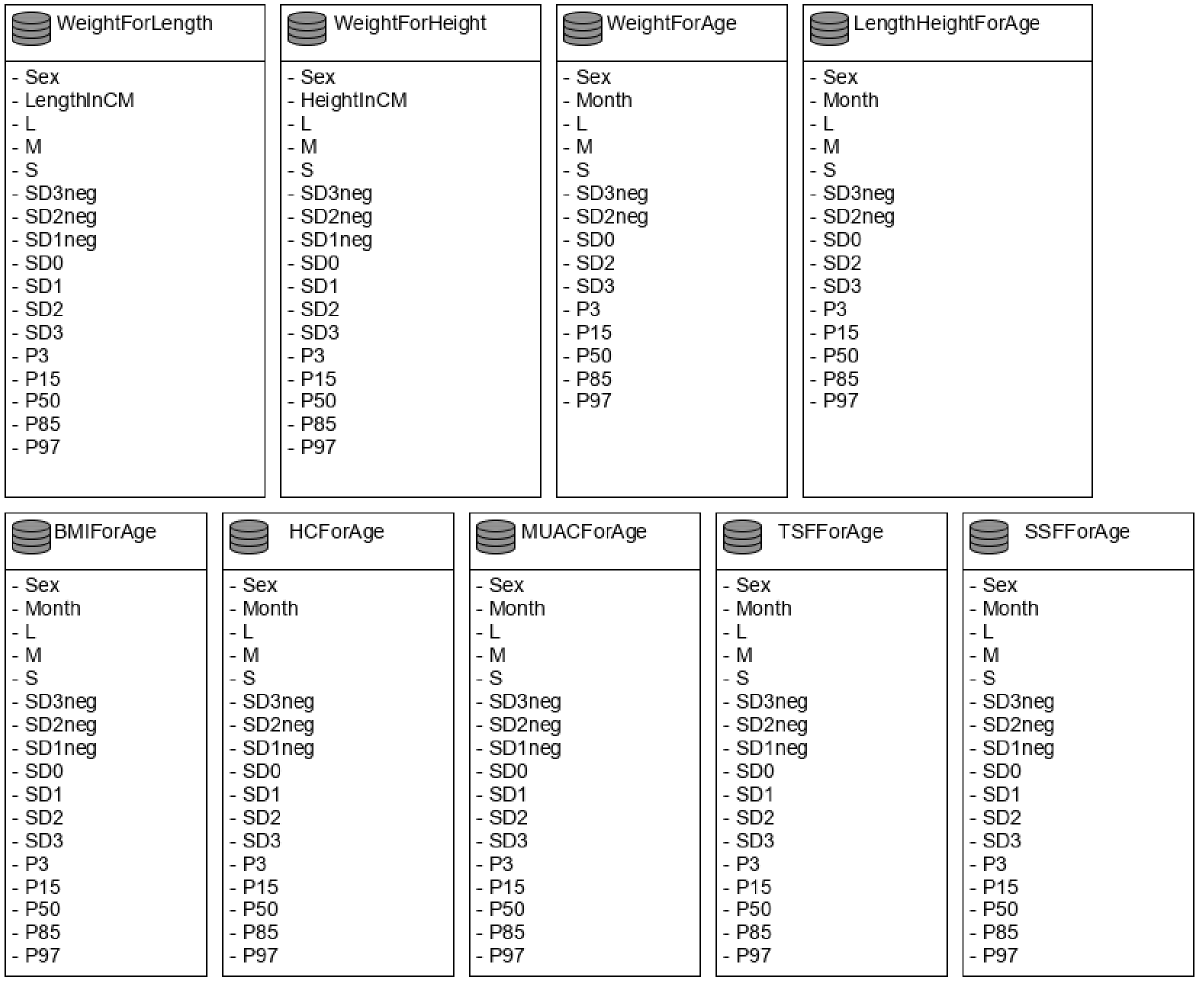


*Figure 6 Class Diagram*

# 5 User Interface Design

The user interface consists of one data entry form and eight graphs.

## 5.1 Data Entry

Figure 7 captures the user interface and associated inputs to necessary to calculate scores and percentiles. Data entry can be performed for the following inputs: date of visit, sex, date of birth, weight, height, measure type, oedema type, head circumference, MUAC, triceps skinfold, and subscapular skinfold.


*Figure 7 User Interface*

## 5.2 Input Elements

Figure 8 show the results of input measurements with range restrictions. The result of data entry outside the range of permitted values will result in on-screen input element validation preventing form submission and calculation.

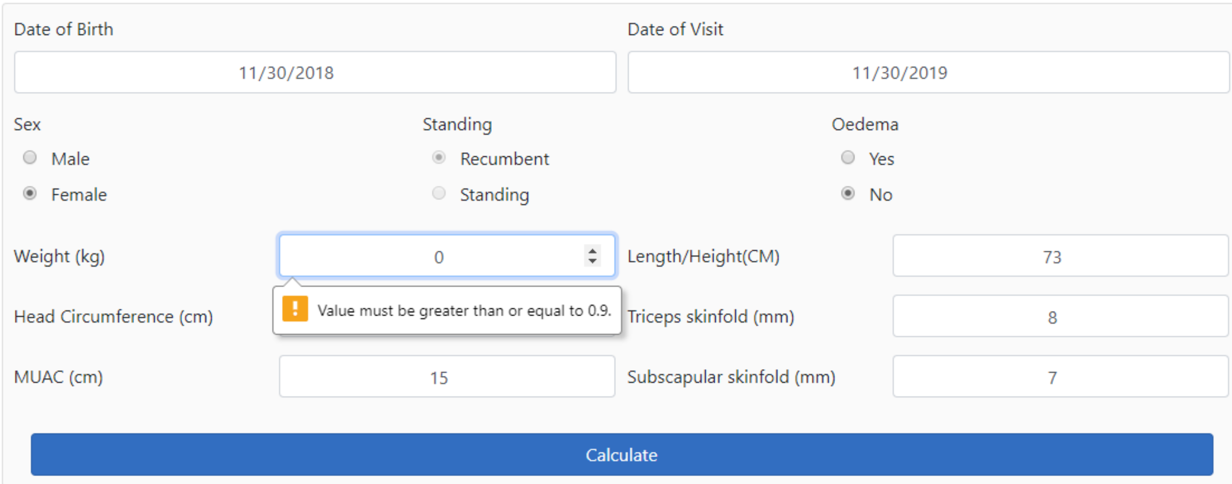


*Figure 8 User Interface Validation*

## 5.3 Chart Elements

The expected chart elements are displayed in the Figure 9 chart. The index is weight-for-height. The chart shows standard deviation growth curves indicative z-scores. The x-axis age measure in completed months and years is displayed. The y-axis weight is measure in kilograms. The patient score is represented as a circle. The growth curves adhere to the four-color scheme. Eight anthropometric chart types can be displayed. The chart format can either be z-score or percentile. The growth standard for all charts will use the WHO growth standard. The x and y-axis will dynamically change per measure. Percentiles will differ in that the curves can be labeled P03, P15, P50, P85, and P97.

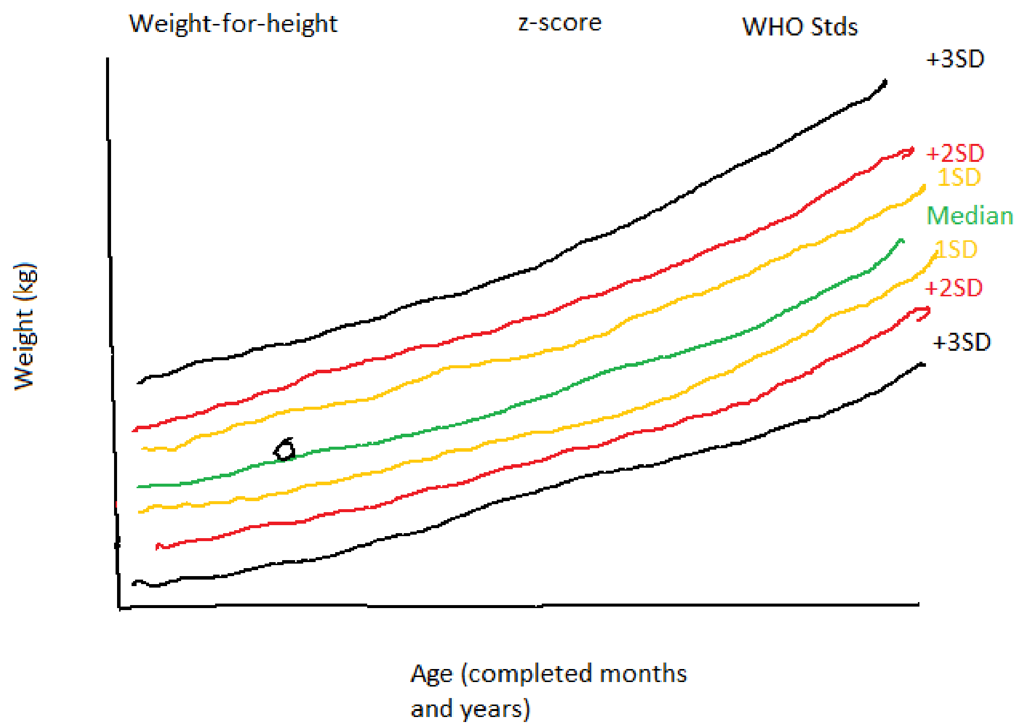


*Figure 9 Chart Mockup*

## 5.4 Color Coding

The color-coded data entry results of z-scores are displayed below. A value between +1 and -1 standard deviation is green. A value between -2 to -1 standard deviation and +1 to +2 standard deviation is gold. A value between -3 to -2 standard deviation and +2 to +3 standard deviation is red. A value outside -3 or 3 is black. The color-coded feedback informs nutritional status. Note: only the background color illustrates color coding in the example.

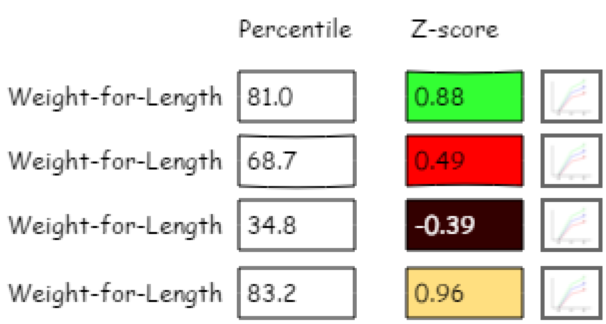


*Figure 10 User Interface Color Coding*

# 6 Requirements Traceability

The requirements established by the Software Require Specification are mapped to the Software Design Document section in the table below.

| Design (SDS) | Requirements (SRS) |
|---|---|
| 3.5.1, 6.2 | UI-1: The solution shall allow data entry for the following inputs: date of visit, sex, date of birth, weight, height, measure type, oedema type, head circumference, MUAC, triceps skinfold, and subscapular skinfold. |
| 6.3 | UII-1: The solution shall follow the input restrictions specified in Table 1. |
| 6.4 | UIG-1: The chart shall have a title indicating the anthropometric index. |
| 6.4 | UIG-2: The chart growth standard used shall be displayed. |
| 6.4 | UIG-3: The chart shall display a title and unit of measure, if applicable, on the x-axis. |
| 6.4 | UIG-4: The chart shall display a title and unit of measure, if applicable, on the y-axis. |
| 6.4 | UIG-5: The chart shall adhere to color coded curve lines. |
| 6.4 | UIG-6: The chart curve will be labeled with the standard deviation or percentile. |
| 6.4 | UIG-7: The plotted score shall be a symbol marker (like an open circle). |
| 6.5 | UIC-1: The solution shall follow the color coding specified in Table 2. |
| 6.2 | UIO-1: The application shall display z-score results on the entry screen for all anthropometric indices. |
| 6.2 | UIO-2: The application shall display percentile results on the entry screen for all anthropometric indices. |
| 6.2 | UIO-3: The application shall display Age and BMI values on the entry screen. |
| 6.4 | UIO-4: The application shall display a chart for all anthropometric indices. |
| 3.3 | HI-1: The solution shall use the Azure cloud platform. |
| 3.3 | HI-2: The solution shall use an Azure App Service. |
| 3.3 | HI-3: The solution shall use an Azure database. |
| 3.3 | SI-1: The solution shall use a Visual Studio integrated development environment. |
| 3.3 | SI-2: The solution shall use an Azure subscription. |
| 3.3 | SI-3: The solution shall use a GitHub repository. |
| 3.3 | CI-1: The service shall use a RESTful API for HTTP requests. |
| 3.5.5 | SF-1: The application service shall calculate Age. |
| 3.5.5 | SF-2: The application service shall calculate Body Mass Index (BMI). |
| 3.5.6 | SF-3: The application service shall calculate Z-scores. |
| 3.5.6 | SF-4: The application service shall calculate Percentiles. |
| 6.2, 6.4, 3.4.1.1 | SF-5: The application shall display a growth chart for eight anthropometric indices. |
| 5.2, 6.4 | SF-6: The application shall chart growth curves for z-scores and percentiles data. |
| 6.2 | SF-7: The application shall chart the individual patient score and percentile. |
| 3.4.2.3, 3.5.9 | SF-8: The database shall store WHO Child Growth Standard data. |
| 3.6 | PER-1: The service shall not exceed 5 seconds for the longest running call. |
| 3.8 | DC-1: All HTML code shall conform to HTML 5.0 standard. |
| 3.3, 3.7 | MNT-1: The solution shall use an application, service, and database. |
| 3.4, 3.7, 3.8 | MNT-2: The solution shall be architected using layers. |
| 3.7, 3.8 | PRT-1: The solution shall use the .NET Core framework. |
| 3.3, 3.7 | PRT-2: PRT-2: The solution shall contain a Dockerfile. |