

CSE 598: Advanced Software Analysis and Design

BMI Calculator - MVC Architecture Mobile App

Purpose:

Software architecture and web data integration are fundamentally important in mobile app design and implementation. This project provides students with the opportunity to put their knowledge of MVC architecture and mobile web integration into practice. Students will develop a simple BMI calculator application using MVC architecture, then use Web APIs to perform the same BMI calculation.

Objectives:

Students will be able to:

- Apply the MVC architecture to design a mobile app that satisfies given requirements
- Implement an MVC architecture-based mobile app
- Use Web API calls
- Process JSON data in a mobile app

Technology Requirements:

Students may develop an iOS app (preferred) or an Android app.

For iOS app:

- XCode 10, programming language Swift (no Objective-C) - **strongly preferred**

For Android app:

- Android SDK using Java

Project Overview:

Design and implement a mobile app that calculates BMI and displays the results to the user. In this app, you use a web API call to calculate BMI and use the API call results to show the BMI to the user.

Project Description:

BMI Calculator Using API:

Create an application that, when given the height and weight of a person, calculates their BMI. In this app, you will use a REST API call to calculate the BMI once height and weight are given as parameters to the API. The API should take the height and weight as parameters and return the BMI, Risk Factor, and Array of web links with BMI information. Your app should read the weight and height and call the API to calculate the BMI. See the example API call in Figure 2 for a height of 60 inches and weight of 156 lbs.

The main UI of your app should look like the following screenshot.

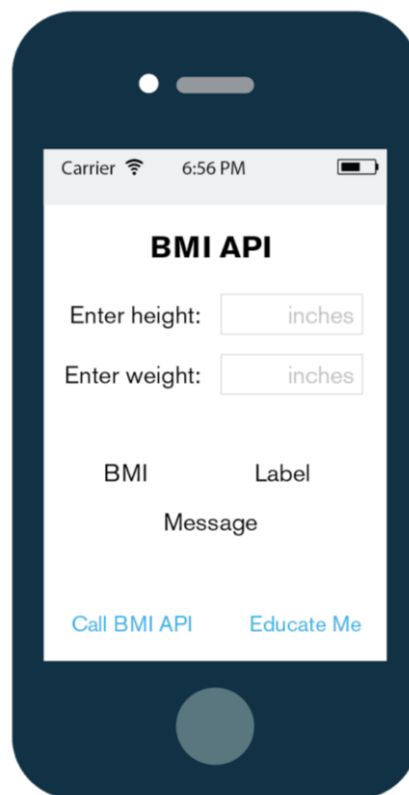


Figure 1: User Interface (UI) Design Guidelines¹

Your application should process the JSON results and display BMI information to the user. Then, displays the following messages to the user based on the BMI value

You are **underweight** if BMI is < 18 – Blue Color You are
normal if BMI is ≥ 18 and < 25 – Green Color

You are **pre-obese** if BMI is between 25 and 30 – Purple

Color You are **obese** if BMI is greater than 30 – Red Color

Additionally, when the user selects the button "Educate Me," the app should load a web page that shows additional information about BMI by using one of the web links (preferably the first web link) from the JSON results. If the "Educate Me" button is clicked before calculating the BMI, the URL with a default value (or without query parameters) can be used to call the API to get the URL and redirect to the web page.

```
http://webstrar99.fulton.asu.edu/page3/Service1.svc/calculateBMI?height=60&weight=156
```

The results from the API call is a JSON document structure:

```
{
  "bmi":30.463333333333335,
  "more":["https://www.cdc.gov/healthyweight/assessing/bmi/index.html","https://www.nhlbi.nih.gov/health/educational/lose_wt/index.htm","https://www.ucsfhealth.org/education/body_mass_index_tool/"],
  "risk":"You are obese :("
}
```

Figure 2: Example API call for a height of 60 inches and weight of 156 lbs

Submission Directions for Project Deliverables:

Submit your project as a separate zip file that implements all the requirements. Name your zip file "**FirstName_LastName_BMI Calculator Project.zip**".

Rubric:

	No Attempt -or- Undeveloped	Developing	Approaching	Meets
Design User Interface Correctly for Phase II	User Interface is missing majority of required UI components based on the specification given	User Interface is missing majority at least half of the required UI components	UI has all the major components, but missing proper alignment, placement in screen	UI has all the required components and has proper alignment and placement in the screen
Make the API Call and Process JSON Data	Incorrect API call and incorrect JSON processing	Correct API call and incorrect JSON processing	Correct API call and few errors (less than 3) in JSON processing	Correct API call and correct JSON processing
Educate Me Button Functionality Implemented Correctly	Educate Me Button functionality not implemented	Educate Me Button functionality incorrectly implemented	N/A	Educate Me Button functionality correctly implemented

¹Images modified from Xcode 10 and originally developed using Swift 4.2.