CSE 460: 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:

Phase I [40 Pts]: Design and implement a mobile app that calculates BMI and displays the results to the user.
Phase II [30 Pts]: Use web API call to calculate BMI and use the API call results to BMI to the user.

Project Description:

Create an application that, when given the height and weight of a person, calculates their BMI in two ways. In the first phase of this project, you will implement a BMI calculator of your own based on the guidelines provided. In the second phase, you will use a REST API call to calculate the BMI once height and weight are given as parameters to the API.

The main view of your app should have two buttons: "Calculate BMI" and "Calculate BMI using API." Each button leads a user to two different views of the app, which implement phases I and II respectively. The overall user interface of the application should follow the guidelines provided in Figure 1.

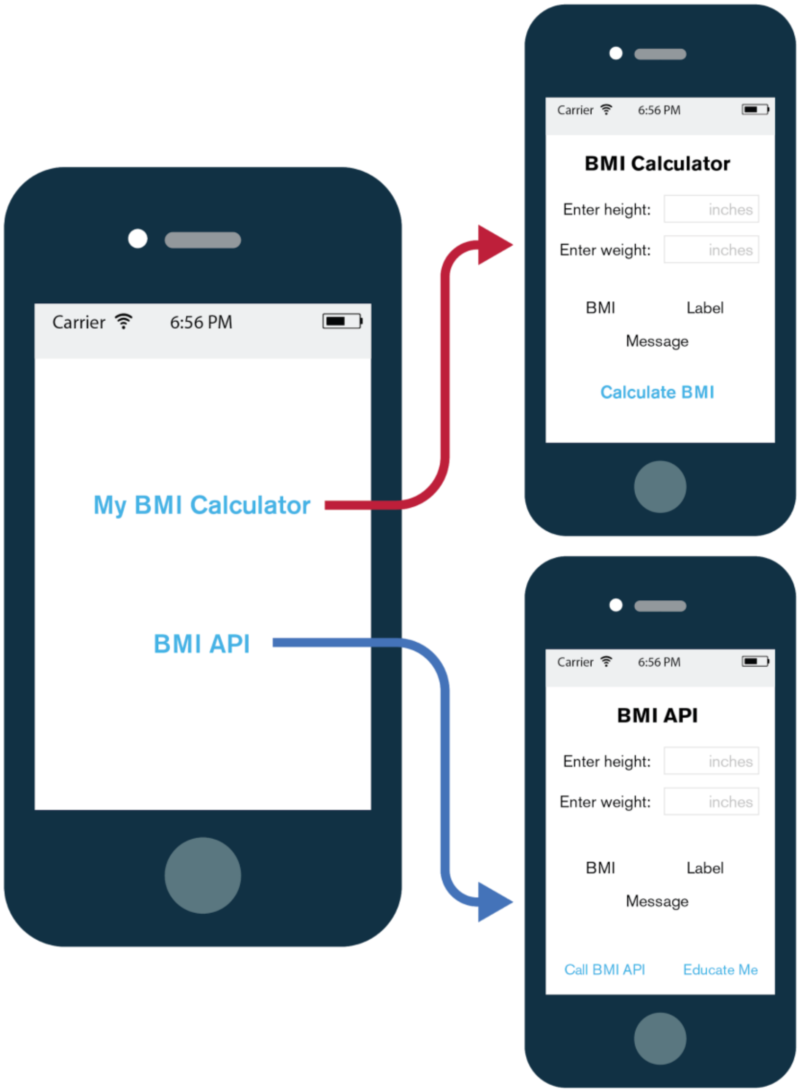


Figure 1: User Interface (UI) Design Guidelines for Phases I and II [1]

Storyboard Design Guidelines for Phases I and II

Note: These design guidelines are based on XCode 10 and developed using swift 4.2. Students using Android SDK can following corresponding design tools available in Android SDK.

Figure 2 shows the storyboard design of the app showing all the views and the file structure required in this application.

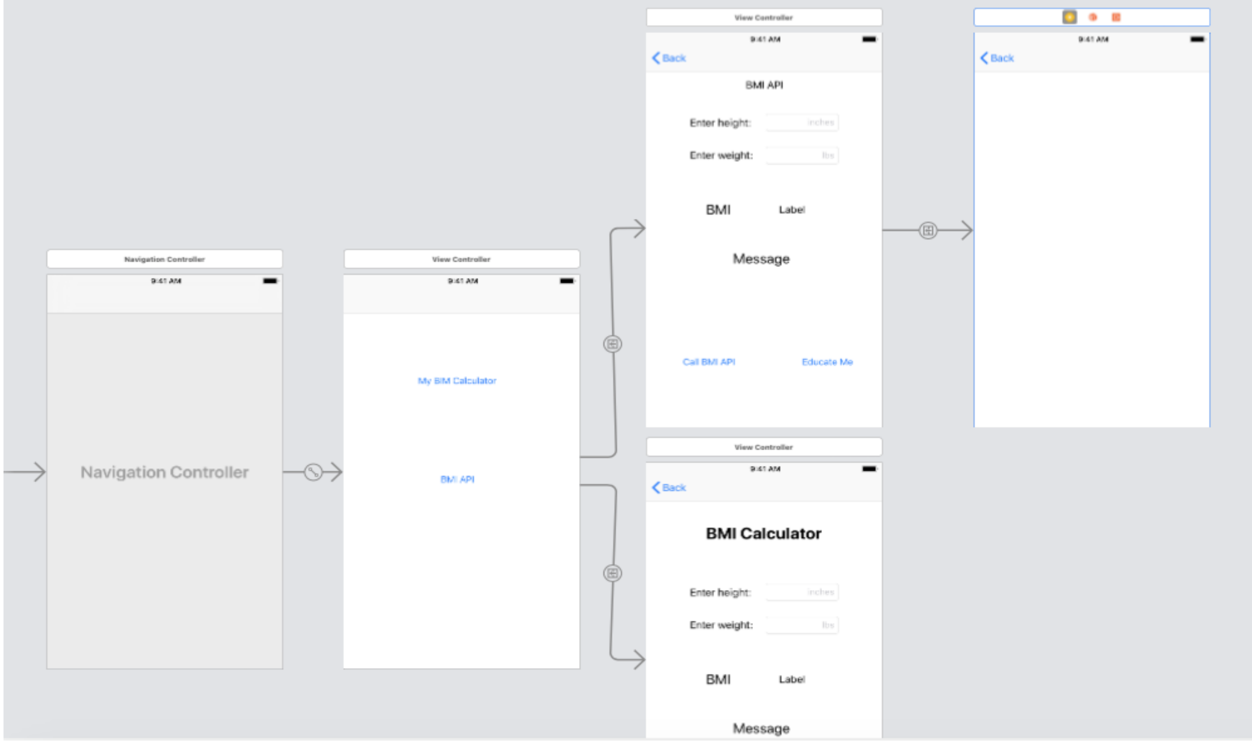
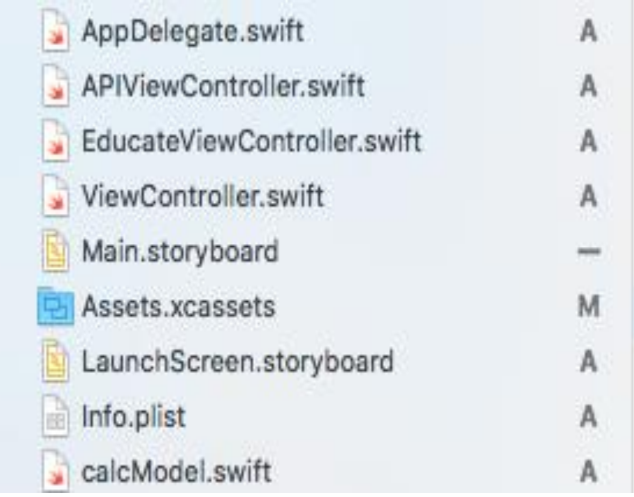


Figure 2: Storyboard Design Guidelines for Phases I and II. [1]



```
AppDelegate.swift            A
APIViewController.swift      A
EducateViewController.swift  A
ViewController.swift         A
Main.storyboard              —
Assets.xcassets              M
LaunchScreen.storyboard      A
Info.plist                   A
calcModel.swift              A
```

Submission Directions for Project Deliverables

Submit your Phase I project as a zip file that implements all Phase I requirements. Name your zip file “FirstName_LastName_BMI Calculator Project_Phase_I.zip”.

Submit your Phase II project as a separate zip file that implements all Phase II requirements. Name your zip file “FirstName_LastName_BMI Calculator Project_Phase_II.zip”.

Phase I:

BMI Calculator

Your application should be designed and implemented using MVC architecture, View and the View controller handles, and UI display and UI actions. The model should perform the BMI calculation and provide the results to the View Controller. See Figure 3 for the UI for Phase I.

Design and implement an application that meets the listed criteria:

- Reads the weight of a person in pounds and height of a person in inches
- Calculates the BMI using following equation

  BMI = (weight in pounds/((height in inches)* (height in inches))*703

- 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

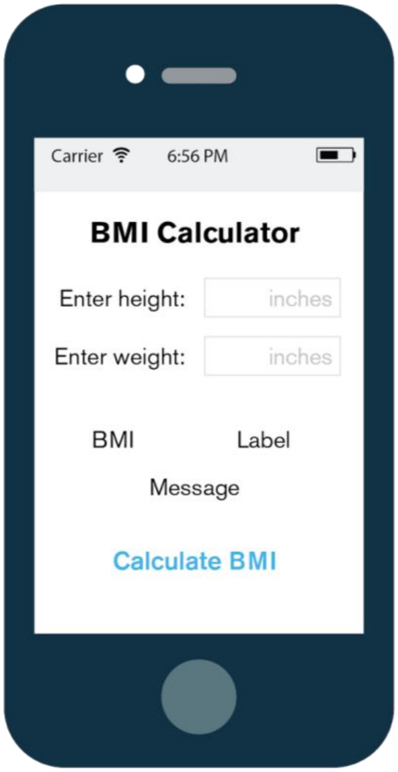


Figure 3: Application UI for Phase I[1]

Phase II:

BMI API

Calculate BMI using a Web API Call. 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 as in Phase I, but call the API to calculate the BMI. See the example API call in Figure 4 for a height of 60 inches and weight of 156 lbs.

Your application should process the JSON results and display BMI information to the user, as in Phase I. 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 from the JSON results.

See Figure 5 for the UI for Phase II.

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

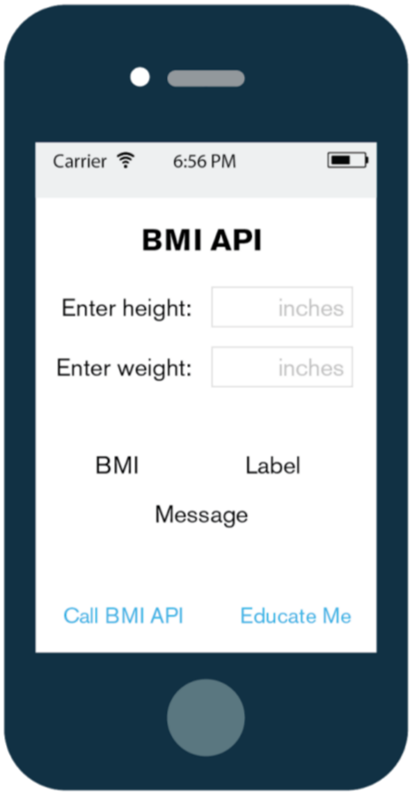


Figure 5: Application UI for Phase II[1]

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