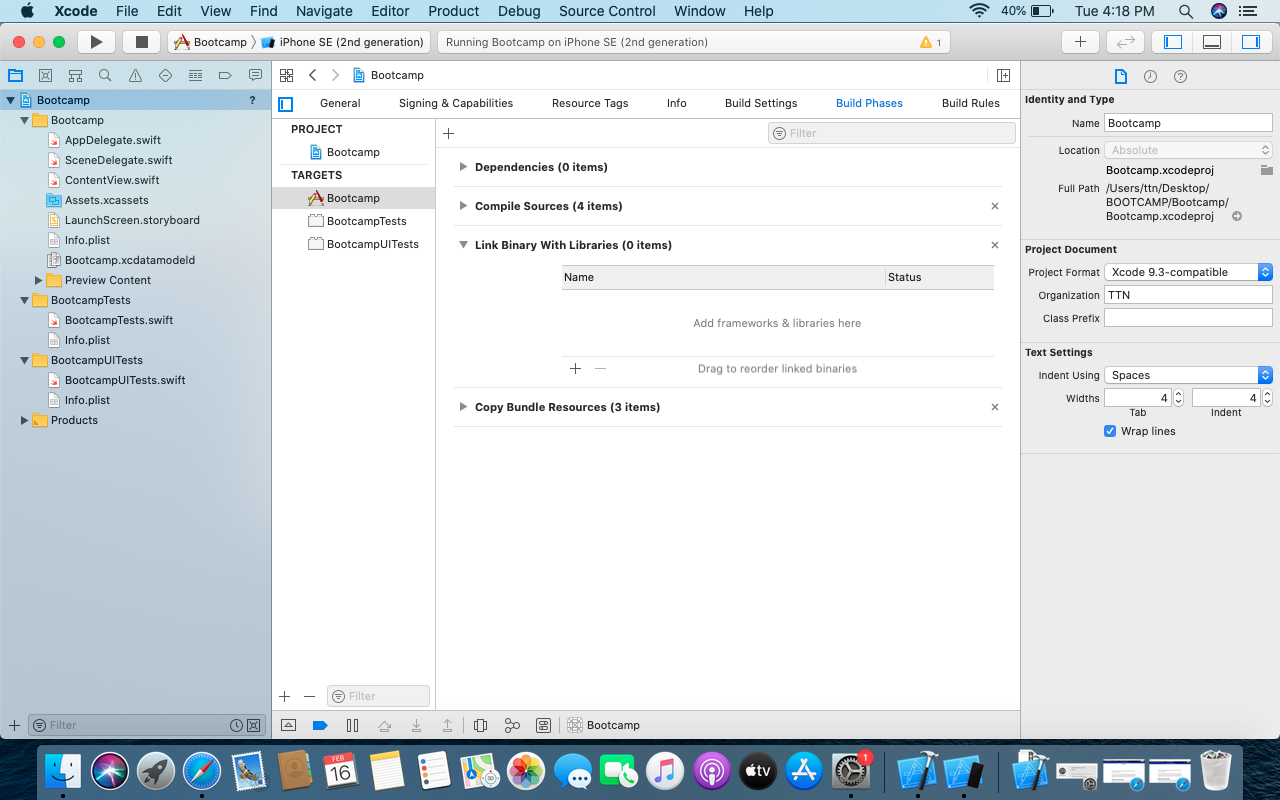
1.

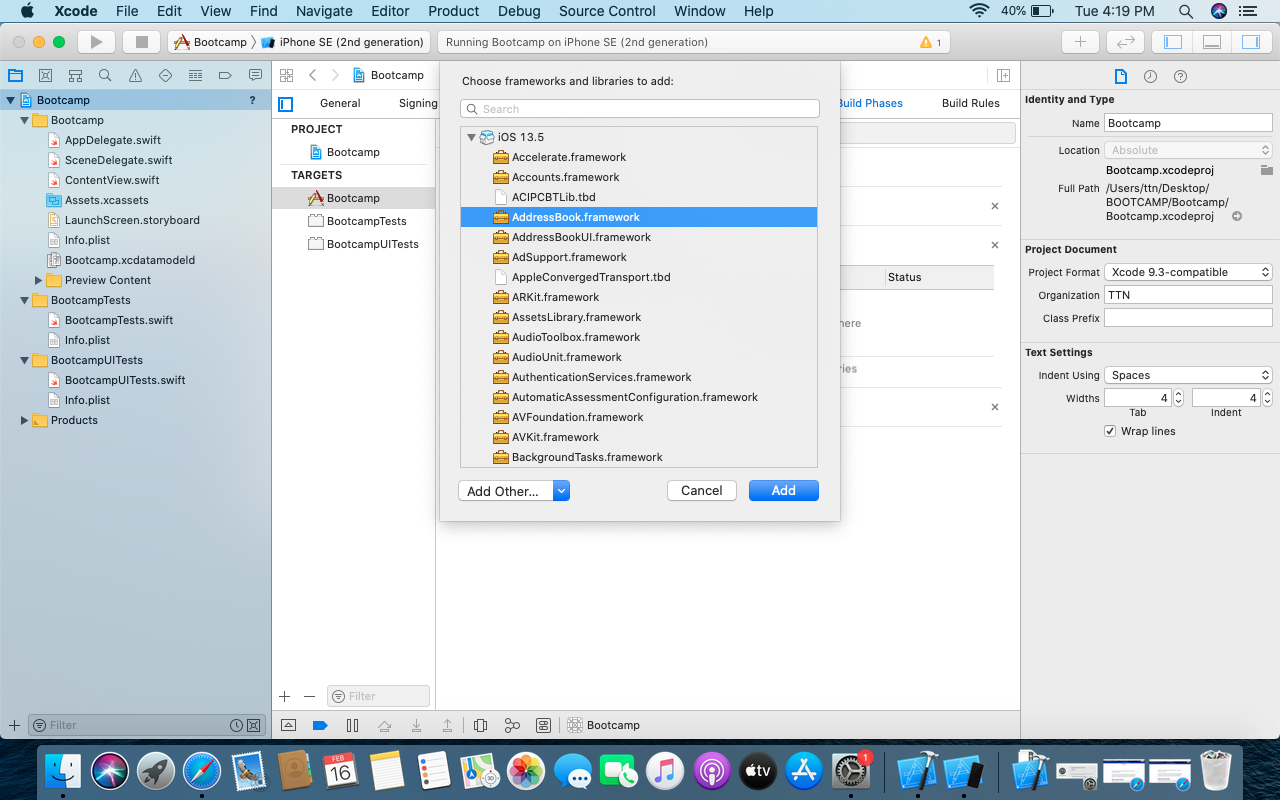
XCode is Apple’s integrated development environment (IDE) that is used to build apps for Apple products including the iPad, iPhone, Apple Watch, and Mac. XCode provides tools to manage an entire development workflow—from creating an app, to testing, optimizing, and submitting it to the App Store.

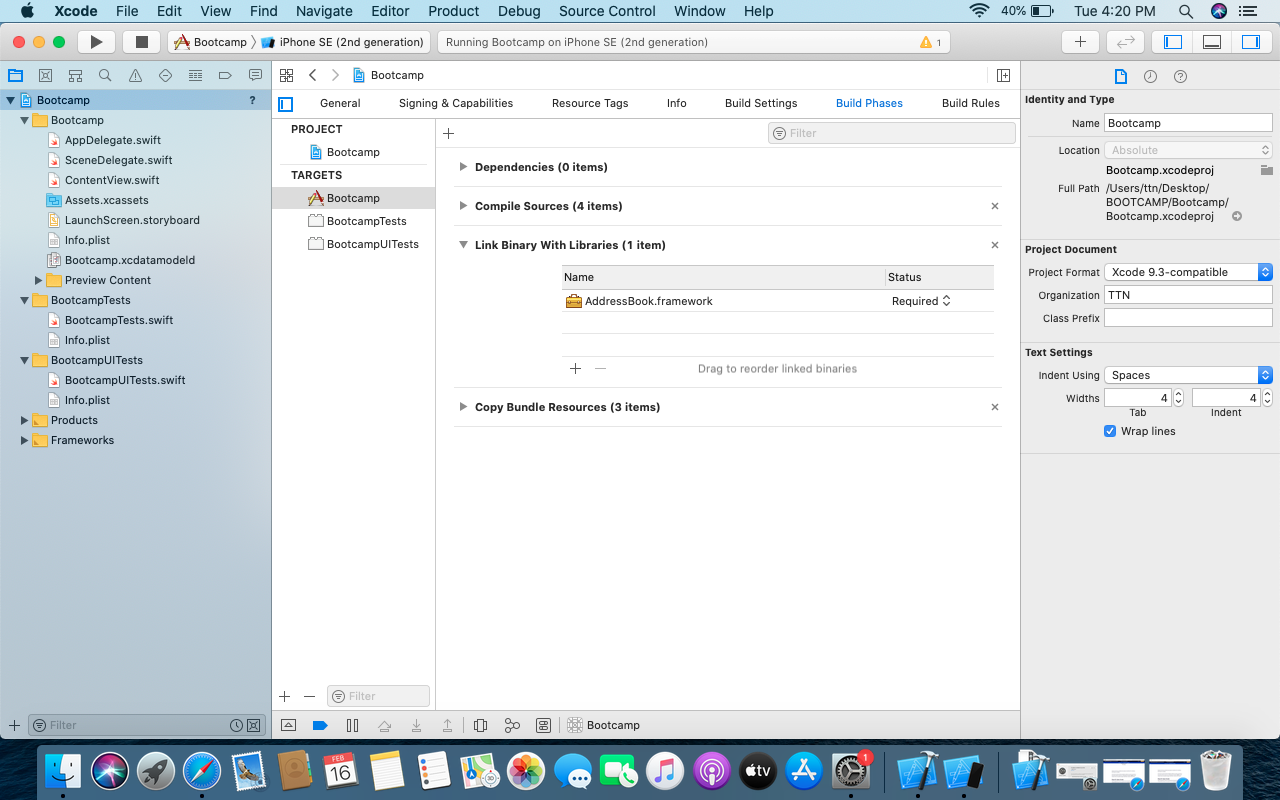
2.

Steps for adding frameworks in XCode project:

1. Select the project file from the project navigator on the left side of the project window.
2. Select the target for where you want to add frameworks in the project settings editor.
3. Select the “Build Phases” tab, and click the small triangle next to “Link Binary with Libraries” to view all of the frameworks in your application.
4. To Add frameworks, click the “+” below the list of frameworks.







3.

**XCode** is Apple’s integrated development environment (IDE) that is used to build apps for Apple products including the iPad, iPhone, Apple Watch, and Mac. XCode provides tools to manage an entire development workflow—from creating an app, to testing, optimizing, and submitting it to the App Store.

**Cocoa** is [Apple](https://en.wikipedia.org/wiki/Apple_Inc.)'s native [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [application programming interface](https://en.wikipedia.org/wiki/Application_programming_interface) (API) for its [desktop](https://en.wikipedia.org/wiki/Desktop_computer) [operating system](https://en.wikipedia.org/wiki/Operating_system) [macOS](https://en.wikipedia.org/wiki/MacOS).Cocoa consists of the [Foundation Kit](https://en.wikipedia.org/wiki/Foundation_Kit), [Application Kit](https://en.wikipedia.org/wiki/Application_Kit), and [Core Data](https://en.wikipedia.org/wiki/Core_Data) frameworks, as included by the Cocoa.h header file, and the libraries and frameworks included by those, such as the C standard library and the Objective-C runtime.

**Objective-C** is a [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [programming language](https://en.wikipedia.org/wiki/Programming_language) that adds [Smalltalk](https://en.wikipedia.org/wiki/Smalltalk)-style [messaging](https://en.wikipedia.org/wiki/Message_passing) to the [C](https://en.wikipedia.org/wiki/C_(programming_language)) programming language. It was the main programming language supported by [Apple](https://en.wikipedia.org/wiki/Apple_Inc.) for [macOS](https://en.wikipedia.org/wiki/MacOS), [iOS](https://en.wikipedia.org/wiki/IOS), and their respective [application programming interfaces](https://en.wikipedia.org/wiki/Application_programming_interface) (APIs), [Cocoa](https://en.wikipedia.org/wiki/Cocoa_(API)) and [Cocoa Touch](https://en.wikipedia.org/wiki/Cocoa_Touch), until the introduction of [Swift](https://en.wikipedia.org/wiki/Swift_(programming_language)) in 2014.

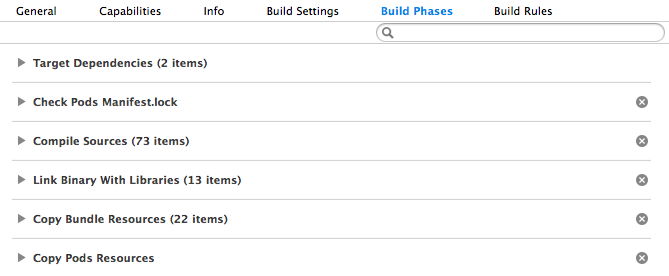
4.

XCode comes with a set of predefined snippets and to open them we can use the shortcut:

Shift-command-L

5.

Build phases represent the high-level plan of how to get from your code to an executable binary. They describe the different kind of tasks that have to be performed along the way.



First, the “target dependencies” are established. These tell the build system which targets have to be built before the build of the current target can commence.

The next step in the process is actually loading the sources. Each .podspec contains a reference to files, normally including a git remote and tag.

The “Compile Sources” section specifies all the files that have to be compiled.

“Link Binary with Libraries” This section lists all static and dynamic libraries that are to be linked with the object files generated by compilation in the previous step.

When linking is done, the last build phase is copying static resources, like images and fonts, into the app bundle.

Although copying static resources is the last build phase, the build process is not complete yet. For example, code signing still has to happen, but that’s not considered to be a build phase.

6.

App delegate is declared as a subclass of UIResponder by XCode project templates.

7.

Constants and variables must be declared before they’re used. We declare constants with the let keyword and variables with the var keyword.

* let maximumNumberOfLoginAttempts = 10
* var currentLoginAttempt = 0

We can declare multiple constants or multiple variables on a single line, separated by commas:

var x = 0.0, y = 0.0, z = 0.0

You can provide a *type annotation* when you declare a constant or variable, to be clear about the kind of values the constant or variable can store.

var welcomeMessage: String

The welcomeMessage variable can now be set to any string value without error:

* welcomeMessage = "Hello"

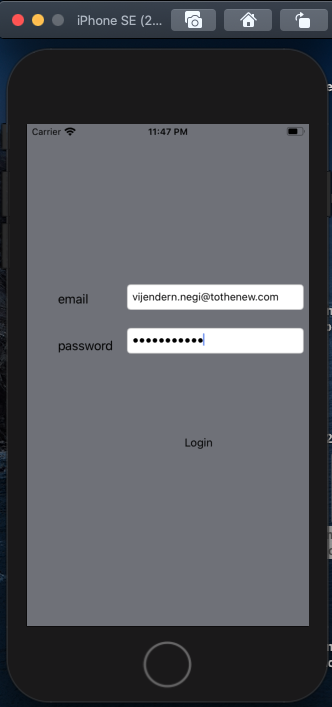
You can define multiple related variables of the same type on a single line, separated by commas, with a single type annotation after the final variable name:

* var red, green, blue: Double

8.

The Interface Builder editor within XCode makes it simple to design a full user interface without writing any code. Simply drag and drop windows, buttons, text fields, and other objects onto the design canvas to create a functioning user interface.

9.



10.

|  |  |  |  |
| --- | --- | --- | --- |
| Device | Resolution | iOS | Display Size |
| iPhone 12 Pro 6.7″ | 2778 X 1284 | iOS 14.1 | 6.7″ |
| iPhone 12 6.1″, iPhone Pro 12 6.1″ | 2532X 1170 | iOS 14.1 | 6.1″ |
| iPhone 12 5.4″ | 2342X 1080 | iOS 14.1 | 5.4″ |
| iPhone 11 Pro Max | 2688 X 1242 | iOS 13.0 | 6.5″ |
| iPhone 11 Pro | 2436X 1125 | iOS 13.0 | 5.8″ |
| iPhone 11 | 1792 X 828 | iOS 13.0 | 6.1″ |
| iPhone XS Max | 2688 X 1242 | iOS 12.0 | 6.5″ |
| iPhone XR | 1792 X 828 | iOS 12.0 | 6.1″ |
| iPhone X, XS | 2436X 1185 | iOS 11.0.1,  iOS 12.0 | 5.8″ |
| iPhone 6+, 6S+, 7+, 8+ | 1920 X 1080 | iOS 9.0,  iOS 10.0,  iOS 11.0 | 5.5″ |
| iPhone 6, 6S, 7, 8 | 1334X 750 | iOS 8.0,  iOS 9.0,  iOS 11.0 | 4.7″ |
| iPhone 5, 6SE5, 5S, 5C, 6SE | 960X 640 | iOS 6.0,  iOS 7.0,  iOS 8.0 | 4.0″ |
| iPhone 44, 4S | 960X 640 | iOS 4.0,  iOS 5.0 | 3.5″ |
| iPhone1st, 2nd & 3rd Generation | 480 X 320 | iOS 1.0,  iOS 2.0,  iOS 3.0 | 3.5″ |