Wristband association App Developer’s Guide

Section 0: Overview 1

Project Credentials 1

User Story: 1

Restrictions and Requirements (MoSCoW method) 1

Reference: 2

Section I Preparing to Develop the app 2

Development IDE 2

BDD Framework 2

Installation: 2

Section 0: Overview

# Project Credentials

contact point: Reza Alemy (author), reza@alemy.net

# User Story:

**As a** Mobile App developer

**I want** to have a boilerplate example of using BDD and MVVM-Light in Xamarin framework

**So that** I can develop multiplatform apps that use the BDD technique for testing and MVVM-light for component management.

# Restrictions and Requirements (MoSCoW method)

* The boilerplate project MUST show how to use Visual Studio to set up the app codebase
* The boilerplate project MUST show how to add Specflow for BDD support to the app codebase
* The boilerplate project MUST show how to write Specflow steps in Xamarin.UITest framework
* The boilerplate project MUST show how to add pages and components using MVVM-Light framework

# Acceptance Tests

Given: The repo is cloned

When: I read the documentation included

Then: I have a step by step explanation on how to setup and develop Xamarin apps using Specflow and MVVM.

Reference and Credits:

Rob Gibbens article for Specflow:

<http://arteksoftware.com/bdd-tests-with-xamarin-uitest-and-specflow/>

Section I Preparing to Develop the app

# Install Development IDE

Selected Technology: Visual Studio, C# language on Xamarin framework

Presented Environment: Visual Studio for Mac

Downloaded from: <https://www.xamarin.com/download>

Version: 7.2.2 (build 11)

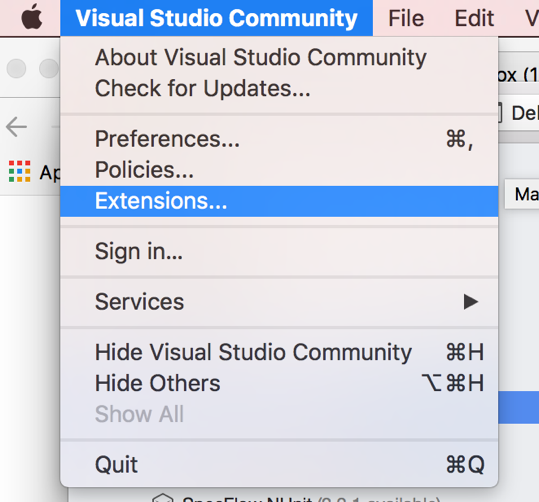
# Install BDD Framework

Selected Technology: Gherkin BDD language

Presented as Visual Studio for Mac Extension: Specflow for mac

Downloaded from: <https://github.com/straighteight/SpecFlow-VS-Mac-Integration/releases/tag/1.11.0.0>

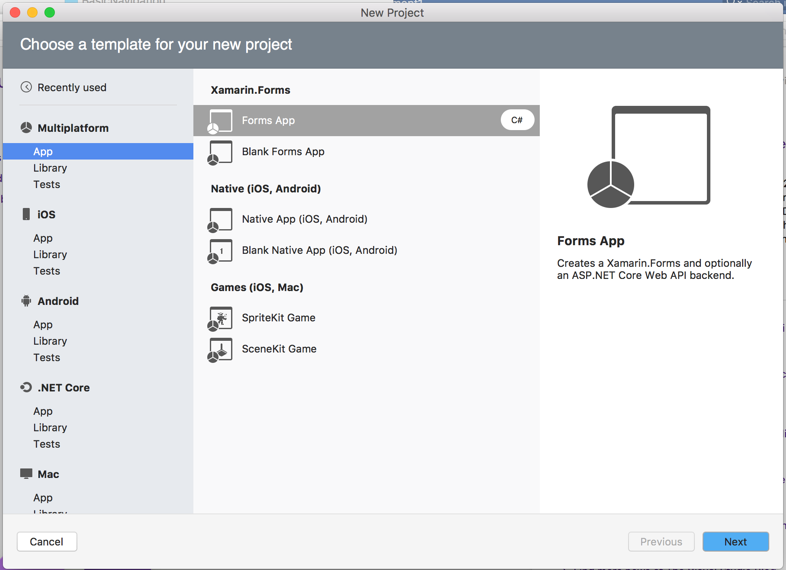
To install, go to 'Visual Studio.../Extensions...' click the button on the far left saying "Install from file..." and select the mpack file from your hard drive. The addin will install and you can now run and creat specflow tests

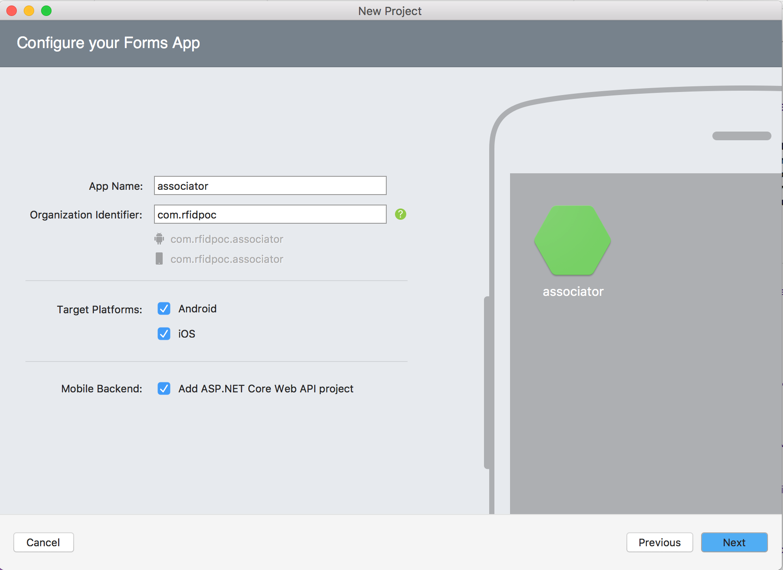


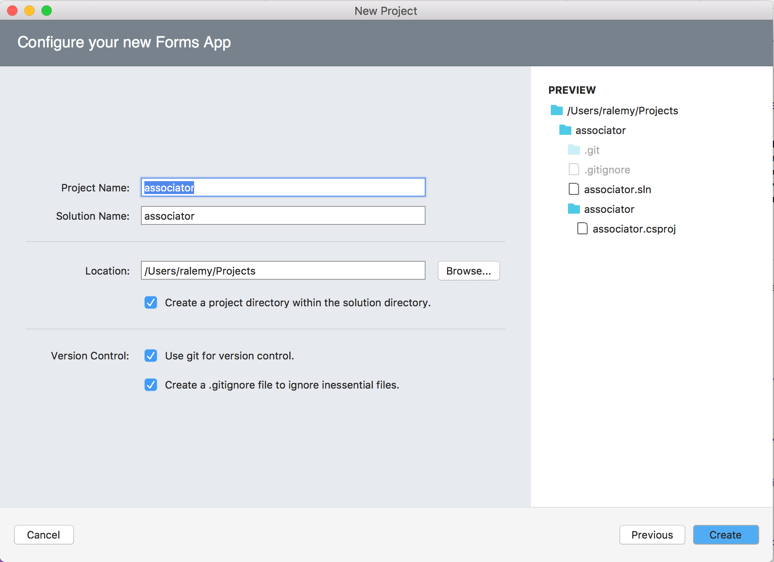
Restart Visual Studio for Mac

# Create Main Solution:

Create a solution with Android and iOS.

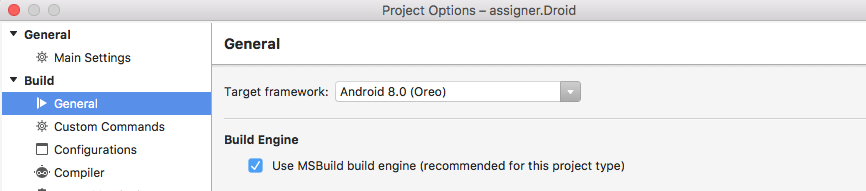




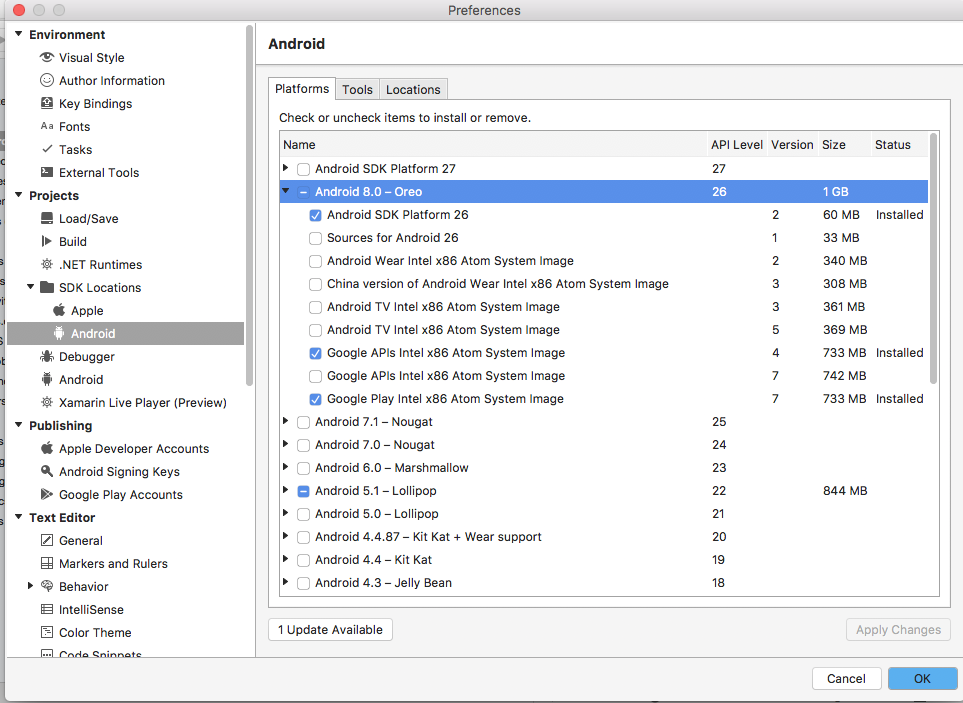


Target the latest Android framework for the Droid project (8.0 for this writing)

* Right click on Droid project and select options
* Go to Build->General->target framework



Install the latest Android SDK for emulator and solution in Tools->SDK Manager



Install the Lollipop Android SDK for emulator

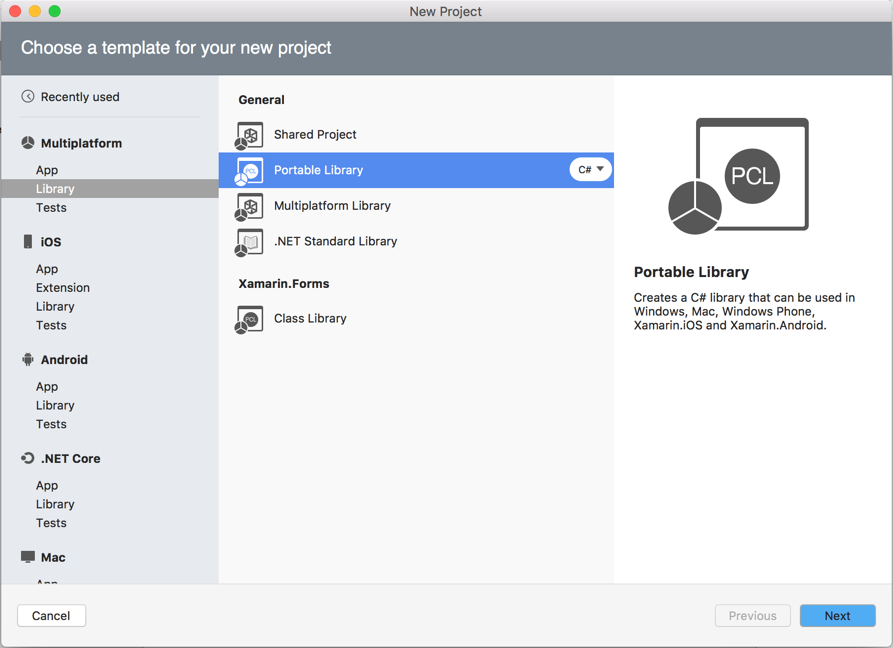
Create a Moto G emulator profile in AVD Manager.

Update Packages for both Droid and iOS.

Add package MVVMLightLibs to both droid and ios frameworks.

# Setup the MVVMLight framework

Add a new project for MVVM framework as a portable library

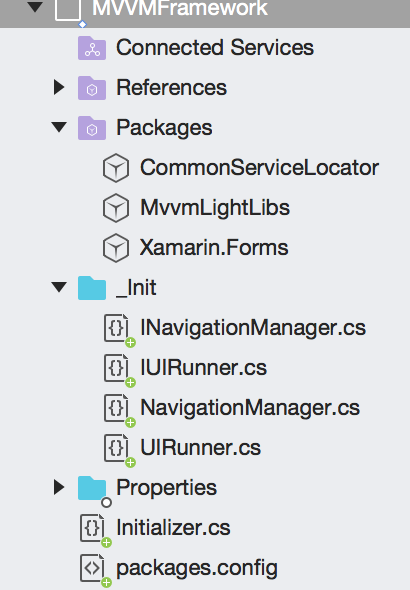


Add MVVMLightLibs and Xamarin.Forms package to this framework

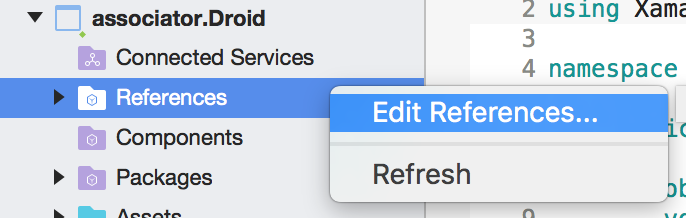
Copy the MVVMFramework/\_init directory from reference repo to this framework

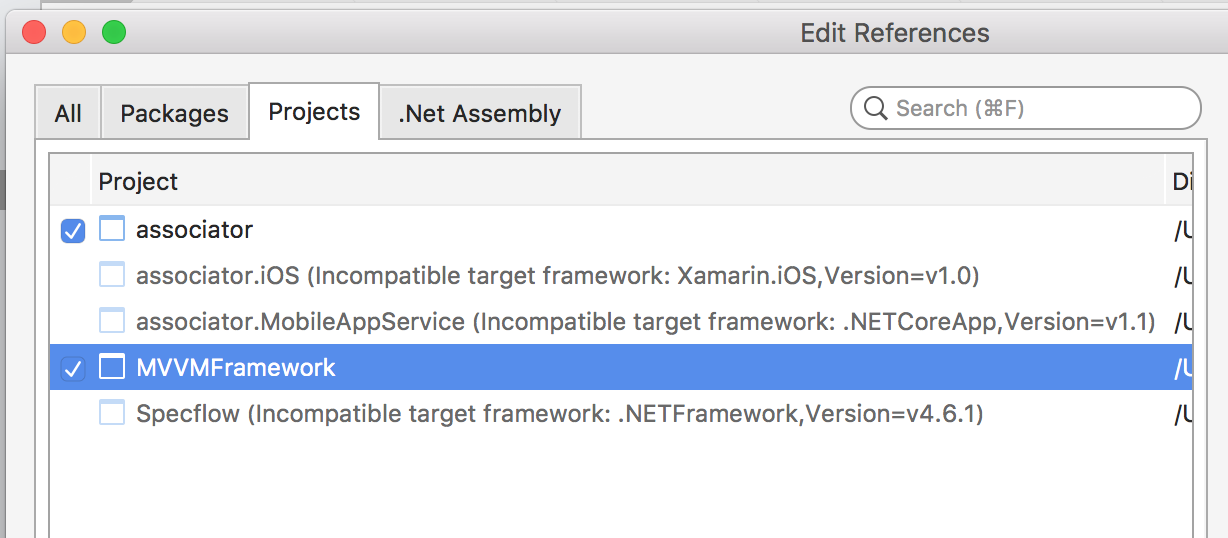
Copy the MVVMFramework/Initializer.cs from reference repo to this framework

Delete MyClass.cs file from the project:



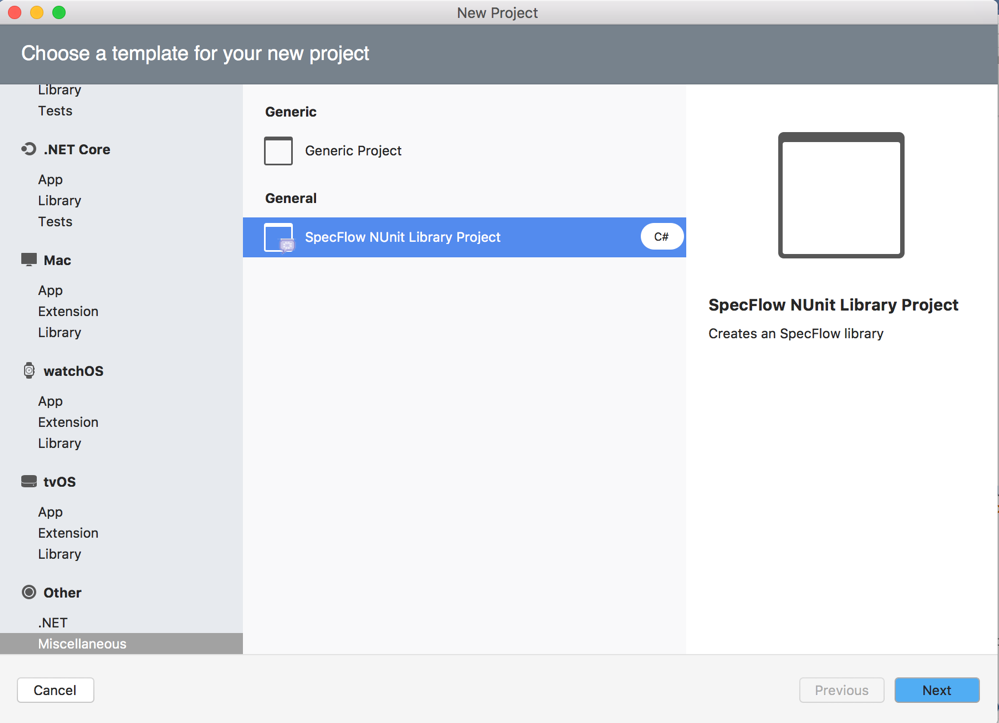
Reference this framework in Droid and iOS projects:

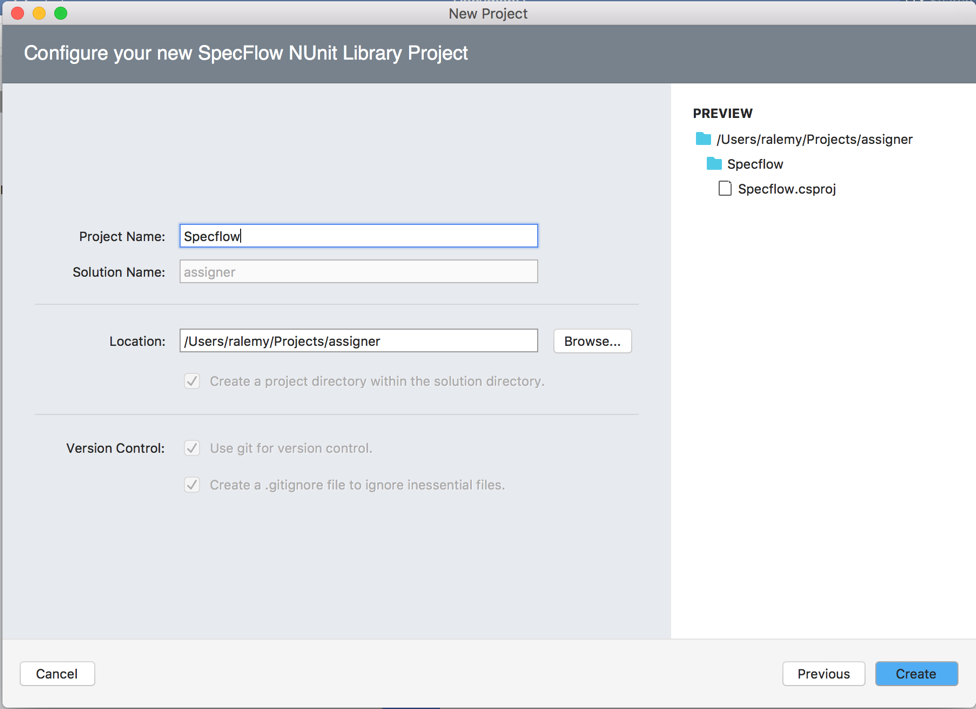




# Add BDD Project

Add a new Specflow project from miscellaneous section



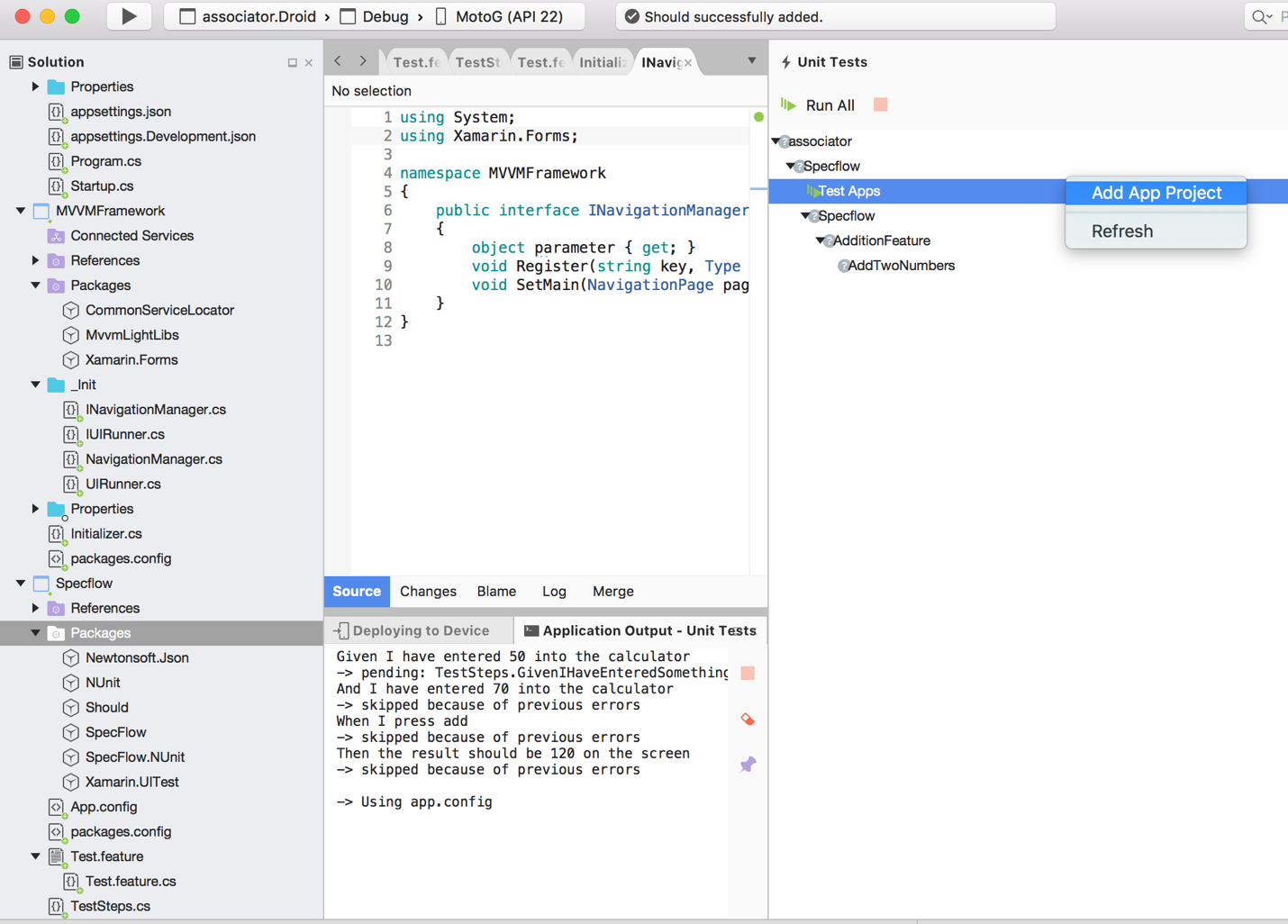


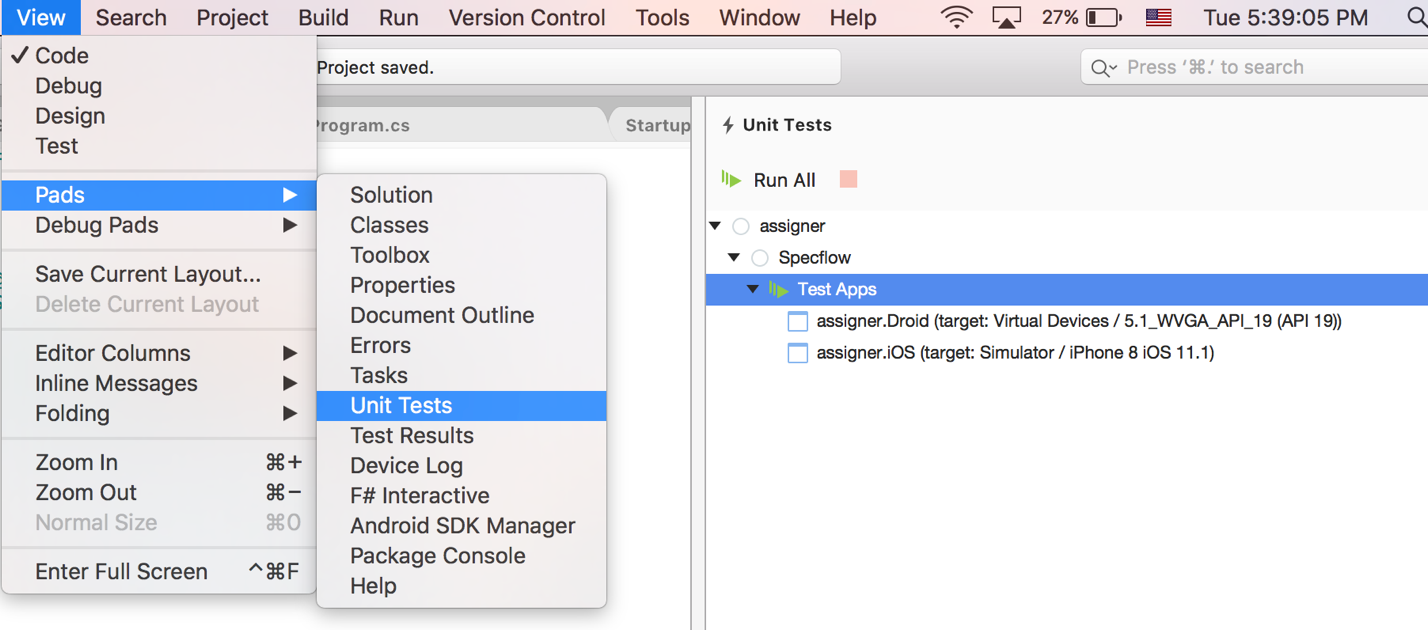
Update packages on Specflow project.

Remove Specflow.NUnit and NUnit packages

Add Xamarin.UiTest, Specflow.NUnit and Should packages to Specflow project.

Open the UnitTest pad from the View menu and add the droid and iOS projects as Test Apps



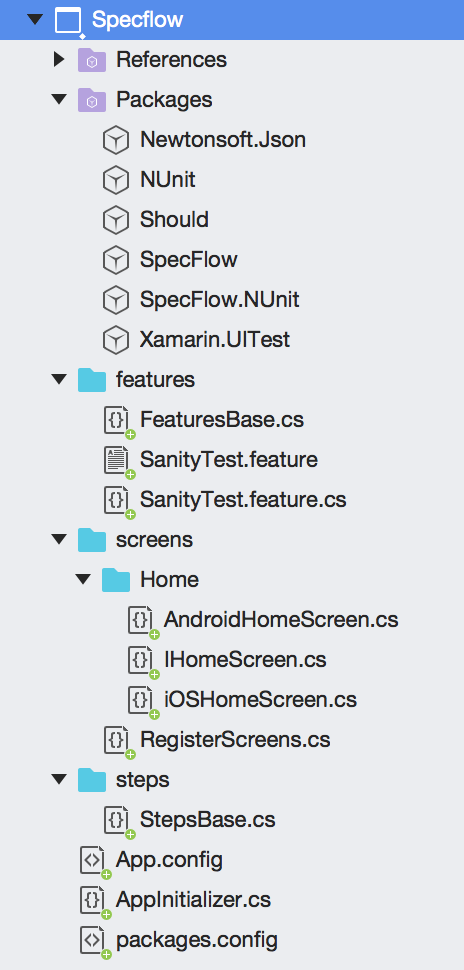


put the AppInitializer.cs in the Specflow project.

Copy features directory from reference repo to Specflow.

Copy screens directory from reference repo to Specflow.

Copy steps directory from reference repo to Specflow.



# Refactoring the Shared Project to MVVM model

From the shared project, remove all folders (Model, View, ViewModel) and files and just keep the App.xaml and App.xaml.cs

Add a Pages and a ViewModels directory.

Add a Context Form page to Pages directory and call it MainPage

Add a Class to ViewModels directory and call it MainPageVM (it has to be a subclass of ViewModelBase)

Replace the contents of the App.xaml.cs to the following:



In the MainPage.xaml.cs, set the binding context to the View model object:



Section 2- Developing the Application

Application is developed following the TDD protocol, i.e. first a test is written against the required behavior, which will fail. Then enough code will be added until the test passes, then the code is refactored to comply with coding standards. The cycle is then repeated, with test followed by code until test passes followed by refactor, until all the requirements and features are implemented.

Start with the first test:

Add a feature file to the features directory of the Specflow project.

Write the first scenario and assign a decorator to it.

Save and rebuild the Specflow project. This will generate the first part of partial class for feature.

Create the second part of partial class for the feature

Run the test

Copy and paste the skeleton code that is given in the error output

Write the test code

Run the test which will fail

git commit

Write code to pass the first test