

## Lab 3 – Saving Activity State

CS235AM, Intermediate Mobile Application Development: Android

This lab will help you understand Android activity life-cycle and give you practice saving activity state. The main concepts you will apply will be:

- Using the `OnSavedInstanceState` callback to store state using a `Bundle` object
- Retrieving activity state in `OnCreate`
- Serializing and deserializing objects using the `XmlSerializer` class
- Using an `EditText` widget

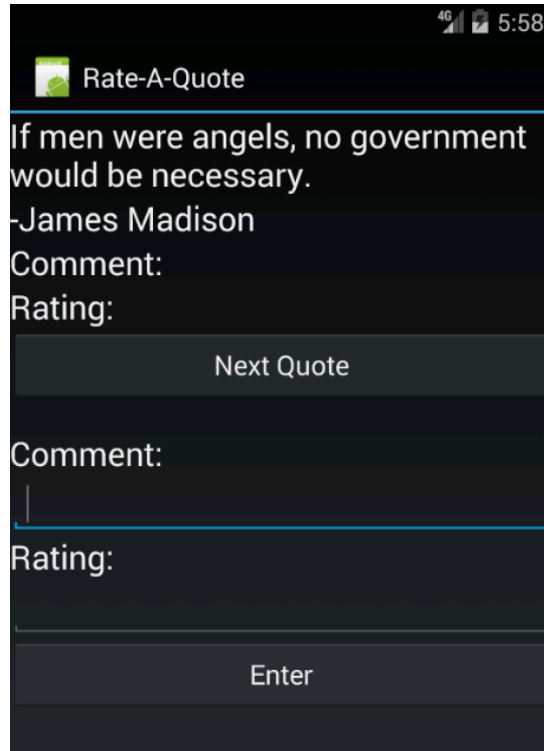
### Part 1 – Tutorial: Activity Lifecycle

Complete the Xamarin walkthrough, [Saving Activity State](#), which is part of the [Activity Lifecycle](#) tutorial, which was also the required reading this week.

### Part 2 for Assignment Group C – Quote Rating App

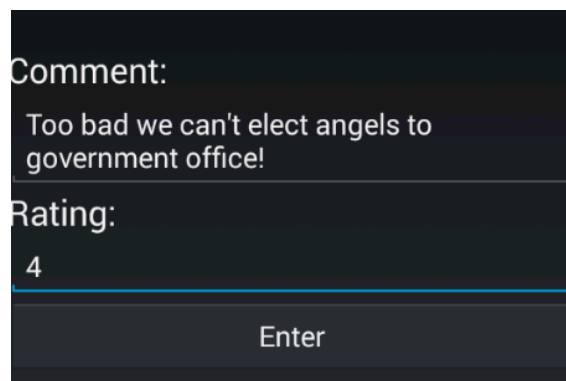
- Make an app that displays quotations.
  - Preload the app with a collection of quotes—at least four.
  - Users will be able to tap a “Next Quote” button see the next quote in the collection.
  - Users will be able to enter comments and a rating.
  - If a quote already has a comment and rating it will be displayed below the quote (You only need to accommodate one comment and rating, not a list of them.)
- Implementation requirements and notes:
  - Use the `XmlSerializer` class to save the object instantiated from the `QuoteBank` class.
  - A file containing the `QuoteBank` class that is preloaded with hard-coded quotes will be provided. You will need to add to it to accommodate comments and ratings.
  - Store the rating and quote in separate properties you add to `QuoteBank`
- Persistence of comments and ratings
  - A quote and rating that has already been entered should continue to be displayed after the device is rotated.
  - The comment and rating should be displayed again after clicking the system back button to quit the activity and then selecting the activity from recent activities and restarting it.
  - The comment and rating should still be shown for this quote if you click the “Next Quote” button repeatedly until it is shown again.
- See screen-shots of the UI below:

The app UI should show something similar to this when it first starts:



The screenshot shows the initial screen of the 'Rate-A-Quote' app. At the top, the status bar displays '4G', signal strength, and the time '5:58'. Below the status bar is the app's title bar with a green Android icon and the text 'Rate-A-Quote'. The main content area displays a quote: 'If men were angels, no government would be necessary.' followed by '-James Madison'. Below the quote are two input fields: 'Comment:' and 'Rating:'. A 'Next Quote' button is positioned between the quote and the input fields. Below the input fields is an 'Enter' button.

A user will be able to enter a quote like this:



The screenshot shows the app after a user has entered a quote. The 'Comment:' field now contains the text 'Too bad we can't elect angels to government office!'. The 'Rating:' field now contains the number '4'. The 'Enter' button remains at the bottom.

And it should look something like this after the user clicks enter:



## Submission to Moodle

### Beta Version

Post the following to the Beta + Code Review Forum:

- 1) For part 1: A document containing screen-shots of the app with each screen-shot labeled. (Please use .docx or .pdf format.)
- 2) For part 2: A zip file containing your app's Visual Studio solution folder. (Make your solution smaller by deleting the *obj* and *bin* folders.)  
Or, optionally, a link to a repository containing your solution source code. You can put the link on the same document with the report on your exercise from part 1.
- 3) A copy of your lab instructions (so the lab partner who reviews your work will know what your requirements were).

### Production Version

1. Items 1 and 2 above, but revised as needed.
2. The code review of your work (the one done by your lab partner) with the second column ("Release") completed by you.