

Peer-graded Assignment: NativeScript UI Elements

1. [Instructions](#)
2. [My submission](#)
3. [Discussions](#)

In this assignment you will explore NativeScript Modal dialog and forms, and use an Action Dialog in your application to initiate actions.

Step-By-Step Assignment Instructions

Assignment Overview

In this assignment, you will use an action dialog in the dishdetail page with two main options and a Cancel option. The first option will enable you to add the dish to your favorites. The second option will bring up a form within a modal dialog that enables you to add your comments about the dish. The Cancel option will dismiss the action dialog.

Assignment Requirements

This assignment consists of two tasks:

Task 1

- Add a Action Item button to the Action Bar with the ios.systemIcon set to 9, and android.systemIcon set to ic_menu_more that when clicked will present an action dialog. You can find more information about Action Dialogs [here](#). Also, you will find more information about configuring the Action Item in the Action Bar [here](#).
- The action dialog should have two main options: "Add to Favorites", and "Add Comment", along with a "Cancel" option.
- Appropriate code is added to Dishdetail page to support the presentation of the action dialog with the options mentioned above.
- When the user clicks on the "Add to Favorites" option in the action dialog, then add the dish to the list of your favorite dishes, by invoking the addToFavorites() function.

Task 2

- When the user clicks on the "Add Comment" option in the action dialog, bring up a modal dialog containing the "form" that enables the user to add comments on the dish. The modal dialog should be created in a folder named Comment, where you add two files comment.component.ts and comment.component.html to create a new component to be used as the modal dialog.
- The "form" should contain three fields: Rating, Author and Comment. Use a [Slider](#) NativeScript component for the Rating, and TextFields for the Author and Comment fields. The Slider should be configured with a range of 1 to 5 with a step size of 1.

- The form should be defined in the template file `comment.component.html`.
- The modal dialog is properly configured within the Dishdetail page so that it can be opened from the DishDetail page.
- Upon submitting the valid comment, the comment should join the other comments for the dish in the Dishdetail component. To do this you need to map the Form model into a data model. The [date](#) for the comment should be set automatically upon submission of the form. You can use the JavaScript [date](#) (see [here](#) for an example) method to set the date to ISO string. Both these actions should be done in the function that handles the form submission in code in the Modal Dialog.
- Thereafter the comment should be returned to the DishDetail component through the `closeCallback()` method. Within `dishdetail.component.ts` file you should configure the code to receive the data within the promise attached to the `showModal()` function.
- You can use the JavaScript array [push](#) method to add the comment into the comments array of the dish.

Review criteria

Your submission will be graded based on the following criteria:

Task 1

- The action dialog is appropriately designed to show the three (Add to Favorites, Add Comment and Cancel) options in the `dishdetail.component.ts` file
- The dish is added correctly to the list of favorites when the user selects the Add to Favorites option

Task 2

- A new component named `CommentComponent` that will act as the modal dialog is added to the application.
- The modal dialog containing the form is shown when the user selects the "Add Comment" option.
- The form is properly configured within the template of the `CommentComponent`
- The reactive form is correctly set up within the `comment.component.ts` file with three fields: author, rating and comments
- The modal is properly retrieving the form data and then adding the current date and passing the resulting comment data to the Dishdetail page through the `closeCallback()` method.
- The Dishdetail component is correctly configured with the `.then()` containing the appropriate callback function attached to the `showModal()` method to receive the data from the modal.
- The comment is properly added to the list of comments for the dish when the user submits the comment.