## PR: Microsoft Web Technologies – Mid-Term Review Exam

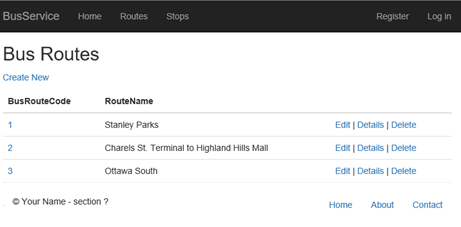
You are working system for a small town’s bus service. Each bus route has multiple stops, and each stop can have multiple routes using it, so there’s a *RouteStop* relation between them. Besides the two foreign keys, each route-stop record contains an *offsetMinutes* field, which denotes how far the given stop is along the given route, in minutes. The starting point for a route is the route-stop record with a zero offset. There can only be one record with a zero offset for each route and a route cannot use the same stop twice.

1. You have an ASP.NET Core MVC project called ***BusService*** and an SQL script called ***BusService.sql*** on the exam account’s G: drive. **Only what is saved to the exam account’s G: drive will be marked.**
   1. Use SQL Server 2017 Management Studio to generate the BusService database from the SQL script.
2. You can work on the project on either the G: drive or the C: drive. If working on the C: drive, **it is your responsibility to periodically copy the project folder to G:**
3. Modify the main layout, ***\_layout.cshtml***:
   1. Add a link to the ***BusRoute*** controller in the menu
   2. Move the *About* and *Contact* menu items to the right side of the footer, aligned horizontally as shown.
   3. Display the TempData variable *message*, bold and red, before the main body of the guest page.
   4. Put your name and section in the footer, as an e-mail hyperlink to your college address.

## BusRoute Controller

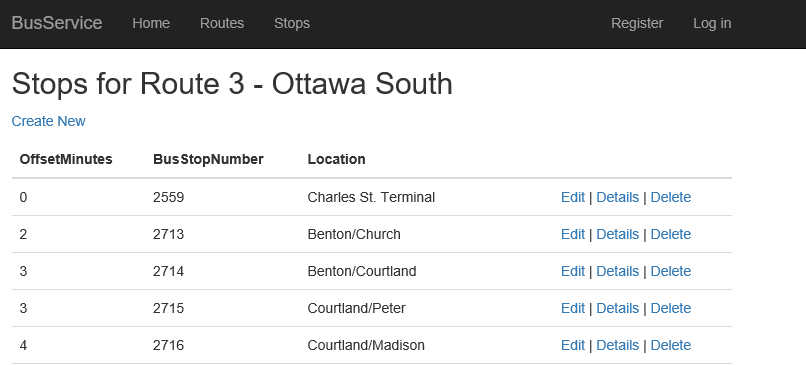
1. Generate the ***BusRoute*** controller with full CRUD support for the *busRoute* table.
   1. Modify its listing of routes to include the busRouteCode.
   2. The page title and main heading should be “Bus Routes”.
   3. Order the Routes by *busRouteCode*
   4. Add a hyperlink to each entry on the listing that passes the *busRouteCode* to the default page of the ***RouteStop*** controller.

This Page intentionally left blank

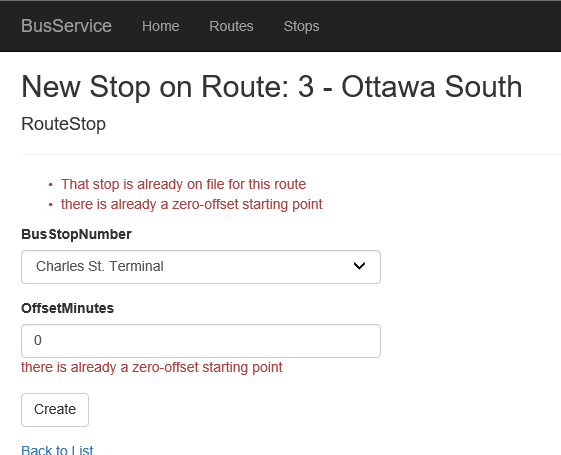


### RouteStopcontroller

1. Generate the ***RouteStop*** controller with full CRUD support for the *routeStop* table.
2. For the *Index* action and its view:
   1. If the *busRouteCode* has been passed in the URL or QueryString, save it to a cookie or session variable
   2. If the *busRouteCode* is not in the QueryString or URL, look for it in a cookie or session variable … if it’s not there either, go back to the BusRoute listing with a message requesting they select a route.
   3. Filter the listing of *RouteStop* records to the given *busRouteCode*, in sequence by offsetMinutes.
   4. The page title & heading should show the route code & name, the listing should show the stop’s number and location, and its offsetMinutes



1. For the Create actions and their view:
   1. Show the location name in the busStop drop-down, sorted by location name
   2. Change the page title and heading to “New Stop on Route: ” followed by the route code and name
      1. Do not show the bus route or bus route code as an input field on the form
   3. When the user clicks save:
      1. The offsetMinutes must be zero or more
      2. There must be one and only one record on the table with offsetMinutes of zero … so the first one added for this route must be zero.
      3. There cannot be a duplicate route/stop combination already on file.
      4. If the record passes these criteria and is successfully added to the database, return to the routeStop listing with a message saying so.
      5. If it does not pass the edits, place a message in ModelState or TempData saying why, and return the model to the Create view.



|  |  |
| --- | --- |
| **\_layout**   * TempData variable displayed before guest page, red & bold * Footer contains student name as an email hyperlink, about & contact links horizontal to right * Menu link sends browser to index page of BusRoute controller |  |
| **BusRoute Listing**   * Page title & heading correct * Listing shows all routes, bus route codes visible & in sequence * Hyperlink on each line redirects user to RouteStop listing   + Link passes busRouteCode |  |
| **RouteStop Listing**   * Check made for *busRouteCode* in URL or a QueryString   + If found, is persisted to a cookie or session variable   + Otherwise, checks for a cookie or session variable with *busRouteCode* & uses that     - If not found, return to route listing & display message to pick an route * Page title & heading correct, includes route code & name * View shows only stops for requested route, sequenced by offsetMinutes * View shows only fields specified |  |
| **RouteStop Create**   * Page title & heading correct, includes route code & name * View shows only fields specified * Stops drop-down shows Location, sorted * On save:   + Model returned to view with pertinent message if:     - No offset zero on file for route & this record is not an offset zero     - This stop is already on file for this route   + Otherwise:     - Missing field(s) added to record     - Record saves to database     - Message displayed on RouteStop listing if insert successful |  |
| **Base Mark**  **Fixes for: build errors run errors @1/fix**  **Final Mark** |  |