# Thursday, September 11, 2014

## Database

1. We need to make one small modification to your database to accommodate future work, so…
2. In SQL Server Management Studio, locate the TicketAttachments table in your database
3. Add a new column named FileUrl which is a nvarchar(MAX) data type and do not allow null entries
4. Save your changes
5. Return to Visual Studio and update your model from the database
6. Save all changes

## Projects

1. Scaffold the Projects class to create a full set of views and controllers
2. Modify the views to make them more user friendly and consistent with the rest of the project
3. We do not need to modify any of the Projects controllers
4. Save and test your project

## ProjectUsers

### Models

1. Create a new ProjectUsersViewModel class that is virtually identical to the UserRolesViewModel – just use ProjectId and ProjectName instead of RoleId and RoleName
2. Add three helper methods to the BTUser class – these are similar to the helper methods you added to the ApplicationUser class, but require a little bit more work:
   1. IsOnProject
   2. AddUserToProject
   3. RemoveUserFromProject
   4. Code these methods according to the comments on the attached page…

### Views

1. Create a new views folder named ProjectUsers
2. Add two new views to this folder (AssignUsers and UsersOnProject), both using the new view model and having NO database context
3. Modify these views as you did with those you created for the UserRolesViewModel (these should look exactly the same, just show project info instead)
4. Add two links to the Projects list view (Projects/Index) as you did for the Roles list view, to assign and view/unassign users to a specific project

### Controllers

1. Using the UserRolesController as a guide, create a new ProjectUsersController with GET and POST methods for the two views you’ve just created.
2. Complete the process of assigning and unassigning users to specified projects, then save and test your program.

## BTUser helper methods

public bool IsOnProject(int id)

{

// create a connection to the BugTracker database

// loop through all entries in the ProjectUsers table

// if the item id is equal to the id passed to this method AND

// the username is equal to this user's username, return true

// return false if we get through the loop without a match

}

public void AddUserToProject(int id)

{

// create a connection to the BugTracker database

// if the this user is NOT on the specified project, add the user

// to the project by creating a new ProjectUser object and adding

// it to the ProjectUsers table, then save the DB changes

}

public void RemoveUserFromProject(int id)

{

// create a connection to the BugTracker database

// if this user is already on the specified project, we need to

// remove the user (see below)...

// loop through all entries in the ProjectUsers table

// if the project id is equal to the id passed to this method AND

// the username is equal to this user's username, remove the item/user

// from the ProjectUsers table, then break out of the loop and save

// the DB changes

}