ApplicationForm refactoring example

1. Instead of inline scripts in .aspx file, scripts should be in separate .js file.

2. Remove all region syntax, because regions can hide “bad” code.

3. ContactPerson class should be in separate file, one class per file.

4. Method PopulateContactPersonList initialize contactPersonList one by one (unmaintainable). Data for ContactPerson should be in database so we can in one line initialize contactPersonList from database with LINQ + Entity Framework/NHibernate.

example:

```
contactPersonList = UnitOfWork.CurrentSession.Query<ContactPerson>().ToList();
```

5. Method GetEmailForMunicipality populates list of contact persons, and then iterate through whole list (bad performance). Instead we can select only Email from database for selected municipality.

example:

```
protected string GetEmailForMunicipality(string municipality)
      ContactPerson person = UnitOfWork.CurrentSession
                              .Query<ContactPerson>()
                              .Where(x => x.Municipality == municipality)
                              .FirstOrDefault();

      if(person != null)
            return person.Email;

      return null;
}
```

6. Method BuildEmailContent builds content of email with HTML tags and it’s not readable. Email template should be defined in separate HTML, and data for email should be passed through model class.

simpler example (EmailTemplate.html):

```
@model Person;
<html>
      <head>
      </head>
      <body>
            <p>
                  Dear @Model.Name, <br/>
                  You won lottery, congratulations.
            </p>
      </body>
</html>
```

7. Exceptions are not logged anywhere (use log4net or something similar).

8. Arguments in BuildMail method are mixed with method call (applicationSender mapped to toAddress, applicationReceiver mapped to fromAddress, applicationReceiver mapped to bccAddress (fromAddress and bccAddress are same in method)). It would be better if we use DTO class as paramether (more maintainable + more readable + OO concept).

```
public class EmailDTO {
      public string Sender { get; set; };
      public string Receiver { get; set; };
      public string BCC { get; set; };
      public string Subject { get; set; };
      public string Content { get; set; };
      public List<Attachment> Attachments { get; set; };
}
```

9. It would be better if validation for subject, email addresses etc are implemented with client and server validation instead of more code in methods.

10. Prefered user language should be read from his user agent information or settings of webpage that he set.

11. MVC approach: Model classes for domain model (classes that represent tables in database) + DTO classes. AJAX in JavaScript which call defined Controller action methods. Use of database + ORM.

12. Much more but this is what I first saw in half an hour.