ApplicationForm

1. Every class should be in separate file: one class per one file (move *ContactPerson* to another file).
2. `BuildEmailContent`: instead of appending strings we should make html file (email template), put all the html in there, add data through model class and pass it through for rendering.
3. Don't use regions: if there's a lot of code in file it is hard for a developer to read it and understand it. Code should be split in several logic units in different files to make understanding and code manipulation easier. It's also easier to detect errors when the code is split.
4. `PopulateContactPersonList`: it's bad practice to initialize data this way. Every time user enters data we should store it in database and then, when needed, fetch it using LINQ.
5. `GetEmailForMunicipality`: use database instead of a list. In existing solution, in order to find one person that matches requested parameters an iteration through the list is needed. Instead, by using LINQ we can return only email like this: person = UnitOfWork.CurrentSession.Query<ContactPerson>().Where(x => x.Municipality == municipality). FirstOrDefault(); and take mail as person.Mail;
6. Define hibernate mapping files for mapping database table and DTO classes for transferring data.
7. There's no exception handling which can lead to unexpected application behavior.
8. Writing inline code should be avoided. Separate js files should be used for script code. Separate css/scss files should be used for style code.
9. Validation should be implemented on both client and server side.