SAVONIA

OPINNÄYTETYÖ - **AMMATTIKORKEAKOULUTUTKINTO**
TEKNIIKAN JA LIIKENTEEN ALA

WEBSITE DEVELOPMENT

Author/s: Nir Graydi

SAVONIA UNIVERSITY OF APPLIED SCIENCES

THESIS
Abstract

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Nir Graydi			
Title of Thesis Website Development			
Date	26 September 2017	Pages/Appendices	29
Supervisor(s) Mr Mikko Pääkkönen, Lecturer Mr Keijo Kuosmanen, Lecturer			
Client Organisation /Partners E-Elämysretket Tmi			

Abstract

The purpose of this thesis was to create a website for registration and activities handling for E-Elämysretket Tmi. This website is the first and only site where customers can register for activities. The company used to use Facebook to publish the activities. With this website, E-Elämysretket can control and check in a real time how many are registered to each activity and have more features. Also, the application fully supports two languages Finnish and English.

Application was done by Visual Studio 2017 Community with ASP.NET Core 2.0, for E-Elämysretket customers and the owner. In this application the owner, admin, has the option to control the whole website. Then in the other hand customers can register to new activities and upload their own experiences and thoughts to the website.

The first part in this thesis was to build the database and layout, which were modifided many times. The second part was to add login and registration with Facebook account. The third part was to code the application. The fourth part was to find a proper hosting company to publish the application. The application was coded by C# and HTML (JavaScript and Bootstrap) and published by Smarterasp.net company.

The result of this thesis was a finalized and finished version of the application, which is ready for use for E-Elämysretket owner and their customers. The application was published successfully to Smarterasp.net hosting company.

Keywords
Asp.net Core HTML Bootstrap JavaScript C#

FOREWORD

I would like to thank Jari Keinänen who gave me this interesting and challenging application. As well, I would like to thank, Mikko Pääkkönen who was my guiding and supervisor teacher.

Kuopio 31 July 2017
Nir Graydi

CONTENTS

TERMS AND MEANINGS

Framework	A framework is like an application programming interface (API) and technically it includes an API. It provides a foundation on which software developers can build programs for a specific platform. (Techterms 2013.)
C#	C# is a multi-paradigm programming language. C# 7.0 version was used in this application, and it came with Visual Studio 2017. (Microsoft 2003.)
Database	A database is an organized collection of data. It is a collection of schemas, tables, queries, reports, views, and other objects. Database designers typically organize the data to their needs. (Microsoft 2006.)
MSSQL	MSSQL Server is a relational database management system (RDBMS) developed by Microsoft. This product is built for the basic function of storing and retrieving data. (Tutorialspoint.com 2017.)
MVC	Model-View-Controller (MVC) is an architectural pattern which separates the application to three components: model, view, controller. In Asp.net MVC is an alternative way to create web application. (Microsoft 2017.)
LINQ	Language Integrated Query is a Microsoft .NET Framework component. LINQ provides shorter and expressive syntax to manipulate data. (Microsoft 2007-02.)

1 INTRODUCTION

The purpose of the thesis will be to create a website for registration and activities handling for E-Elämysretket Tmi. Two login options will be created for this application, one for customers and one for admin who is the owner (E-Elämysretket).

For E-Elämysretket, this is their first company website, which is important for the company. Before this website, E-Elämysretket was handling activities registration by phone, email, short message service (SMS) and Facebook (one of the biggest online social networking services). E-Elämysretket will use this application as their official website. The company will have more features and it will be able to control and check in a real time how many customers are registered to each activity.

The owner will be able to add, delete, edit, upload pictures, publish activities, catch fish on the fish-page and send emails concerning new upcoming activities. All users who will register to the system will be controlled by the owner.

Activities and fish will be browseable. Also, the registration and login are going to be made by Facebook. Fish will be added and edited with wanted pictures. Comments, replies and the email notification system will be viewed and if needed, changed in everyone's own profile page.

Globalization and localization will be supported by this application. Two languages, Finnish and English will be used in this application. More languages will be able to be added in the future.

The application will be made with Visual Studio 2017 Community with Asp.net Core 2.0 framework. ASP.NET Core is a cross-platform, high-performance, open-source framework for building modern, cloud-based, Internet-connected applications. Asp.net core application will be able to be hosted in systems like Windows and some Linux versions. (Microsoft 2017.)

Developing with the Core framework and adding the registration and login with Facebook will require a lot of studying before starting to work on this application. C#, HTML and JavaScript will be the main languages used in the application.

Last stage in this thesis will be to find a proper hosting company which will support all the required features in this application. Features, frameworks and code examples will be shown and explained in this report.

2 FRAMEWORKS AND FEATURES

2.1 Visual Studio 2017 15.3.3 Community

This application was coded with Visual Studio 2017 15.3.3 Community. Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft.

Visual studio is used to develop computer programs for Microsoft Windows, as well as web sites, web apps, web services and mobile apps. Visual Studio 2017 Community is free version and available for everyone to download and use. (Visual Studio Microsoft 2017.)

2.2 Asp.net Core 2.0

ASP.NET Core 2.0 is the framework which was used in this application. It includes Razor Pages user-interface design. Core framework uses models, controllers and views as basic MVC. (Microsoft 2017.)

There are two kinds of Asp.net frameworks. Full .Net framework and .Net Core. .Net Core is the clean version, which means, all extensions were needed to be added one by one. In the other hand, full .net framework, the extensions are already added to the application.

2.3 Facebook developing

Facebook development was used in this application to login and register to the website with Facebook account. App ID key and secret code are required for developing with Facebook and to gain an access to Facebook features. (Facebook 2017.)

Using the Facebook feature in this application, helps users to login by one click to their account, without any email address or password. Facebook was one of requirements in the application and was chosen because it is one of the biggest social network websites and most of the customers have a Facebook account.

2.4 MailKit

MailKit is an Open Source cross-platform .NET mail-client library that is based on MimeKit. It was used in the application to send HTML emails to the customers. (MailKit 2017.)

Emails with pictures and links are possible to be sent with MailKit. The layout of the email message with HTML components were needed to be hardcoded.

2.5 CoreCompat

Pictures were resized with a very good quality by CoreComp.System.Drawing library. It has a MIT License, which was acceptable to use in the application.

It was possible to resize to the needed width and length for each uploaded picture. Resizing the pictures lowered the size (bytes) of the picture.

2.6 Bootstrap

Bootstrap supports responsive web design. This means the layout of web pages adjusts dynamically, considering the size of the used device (desktop, tablet, mobile phone). (Bootstrap 2017.)

Bootstrap was used in the application by the default template of Asp.net Core. These days' customers use a mobile device for almost everything, which makes Bootstrap very useful.

2.7 System.IO

System.IO is a library in .Net Framework. This library enabled to copy and delete files from any device to the server. (Microsoft 2017.)

There was a limitation when uploading pictures to the server. Jpeg, png and jps were the allowed types. Other limitations were total size of 20MB or five images per one upload.

3 PROCESS

3.1 Database structure

MSSQL type of database was chosen for this application and it was the default type of Asp.net Core. Website's data information was stored in the database. The application included the individual user accounts database that was created by Visual Studio. Database's tables are shown in Image 1.

The database has the following tables:

- Blog (Business information): BlogID, BlogName, BlogUrl, CreatorName, Description, Title, Email.
- Posts (Activities): ID, CreatedDate, Description, IsPublished, Meta, GoOnDate, Title, City, PlaceName, ImagePic, ExtraImage, ExtraInfo.
- Replies: ID, Body, CommentID, ReplyDateTime, Deleted, UserName, UserAvatar.
- Participants: ID, PostID, UserID, UserEmail, UserName.
- Comments: ID, Body, Country, PostedDateTime, IPAddress, IsApproved, IsSpam, PostID, UserAvatar, UserName, IsReplied, UserID.
- FishDB: ID, CreatedDate, FishImages, PersonTakenName, PostID, FishName, TrapName, Place, Weight, Length, PicTakenPerson, UserID.
- Notifications: ID, AddedDate, Email, UserFirstName, UserID, UserLastName.

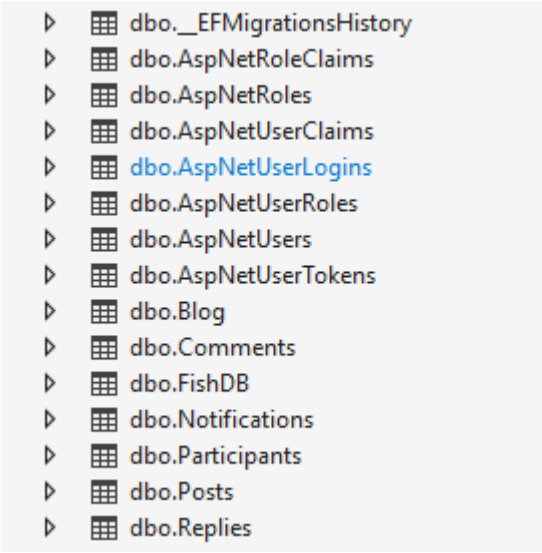

IMAGE 1. Tables used in this application.

3.2 Globalization and localization

The resources folder is used for switching between the languages, Finnish and English. By adding the necessary code in the Startup.cs class in the application, it is possible to use those files to switch between languages on the website.

There are many ways how to write the resources files. One of the ways which was used in this application is explained here:
In case that the view's name is ContactUs.cshtml, then the resource file's name can be Views.Account.ContactUs.fi for Finnish and Views.Account.ContactUs.en for English.

Globalization and localization were needed to set up inside the Startup class (Microsoft 2017). Localization need to be used inside a controller as well. Examples of using the localization inside the controller and the view are shown in Images 2 and 3.

```
while (valid)
{
    if (files != null)
    {
        if (files.Count > 5)
        {
            return _localizer["ErrorMaxFiles"];
        }
```

IMAGE 2. Example of using Localizer in a controller.

```
@using Microsoft.AspNetCore.Mvc.Localization
@inject IViewLocalizer Localizer

@{
    ViewData["Title"] = Localizer["Title"];
```

IMAGE 3. Example of Razor Pages user-interface using the Localizer.

3.3 Models

Models helped to create the columns in the database, error messages and more options. In this application, each table in the database was created from a model class. Model was included the type of every parameter and error message. A blog, comments, participants, posts, replies and fishdb models were added to this application.

There are many ways to create a database in asp.net core with an Entity Framework Core (Entity Framework Core is an object-relational mapper (O/RM) that enables .NET developers to work with a database using .NET objects) (Microsoft 2017). One option is, first to create the database with tables, then from the database, the models are created. Another way is, first to create the model and the database with tables is created from the model.

This application, the database was created from the models (an example of a code of a model is shown in Image 4.). After creating the model, it was needed to run "Add-Migration ModelName" commend in Package Manager Console to scaffold a migration to create the initial set of tables for the model. Last, was needed to apply the new migration into the database with "Update-Database" comment. (Microsoft 2017.)

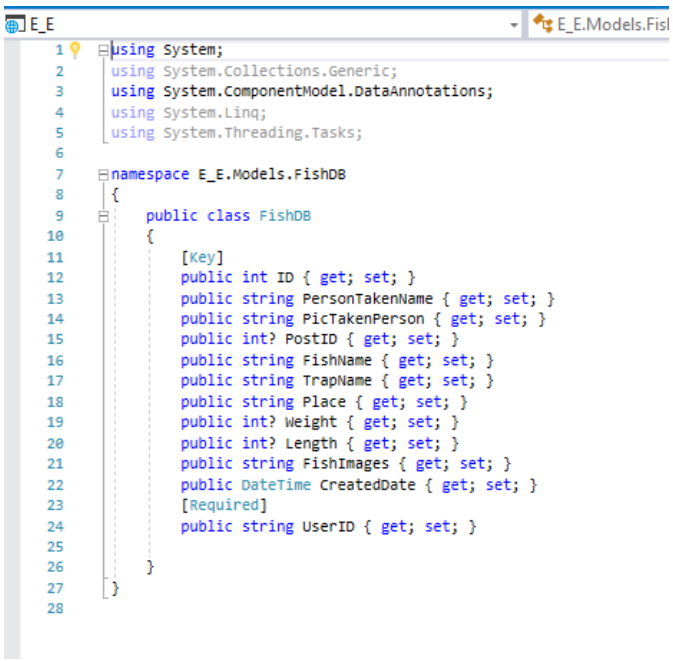

IMAGE 4. Example of fishDB model class.

3.4 Controllers and Views

View was the user interface (in this case, HTML) and displayed the model to the user. The controller contains the flow control logic. A controller determines what response to send back to a user when a user made a browser request. (Microsoft 2008-08-19.)

For every model, there was an option in the controller and in the view to create, delete, details, edit and to show information, the index, as shown in the Image 5. These views were used to add information to the database thru the view by the user. View bags and view data were used to move information between the controller and view.

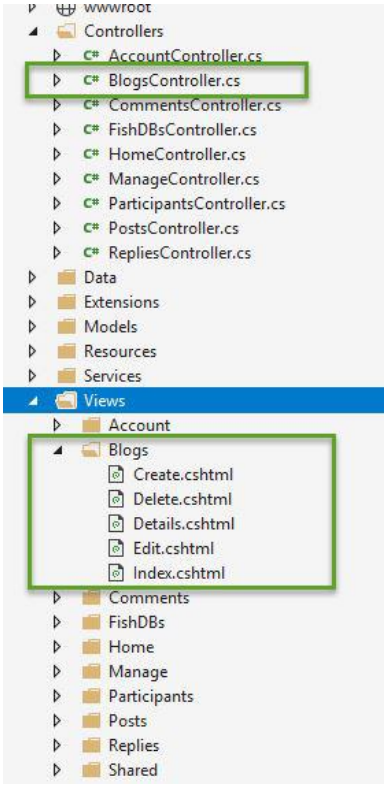

IMAGE 5. Structure of the code, controllers and views.

3.5 Startup.cs

Startup.cs was an important class in this work and was included necessary setup codes. The name of this class can be modified to any name and the default name was Startup.

Startup.cs class checked if Admin information was existed in the system. If not, it will create it by its self. Furthermore, the Startup class was included the resource location for switching between languages, editing the password type and adding Facebook authentication information.

3.6 Services

The Services folder was included the EmailSender class, which is used to send emails to the customers. This class uses the MailKit package, which helps setting up the Simple Mail Transfer Protocol (SMTP) port number, port name, email address to whom and from and the HTML message itself. A code example of this class that shows how to send an email shown in Image 6.

```
try
{
    MimeMessage message = new MimeMessage();
    message.From.Add(new MailboxAddress("Joey Tribbiani", "YOU_FROM_ADDRESS@gmail.com"));
    message.To.Add(new MailboxAddress("Mrs. Chanandler Bong", "YOU_TO_ADDRESS@gmail.com"));
    message.Subject = "How you doin'?";

    message.Body = new TextPart("plain")
    {
        Text = @"Hey Chandler,I just wanted to let you know that Monica and I were going to go play some paintball, you in?-- Joey"
    };


    using (var client = new SmtpClient())
    {
        // Note: since we don't have an OAuth2 token, disable
        // the XOAUTH2 authentication mechanism.
        client.AuthenticationMechanisms.Remove("XOAUTH2");
        // Note: only needed if the SMTP server requires authentication
        client.Authenticate("YOUR_GMAIL_NAME", "YOUR_PASSWORD");
        client.Send(message);
        client.Disconnect(true);

    }
}
catch (Exception ex)
{
    throw ex;
}
return Task.FromResult(0);
```

IMAGE 6. The example code is from the stackoverflow.com which published as a shared example. (Stackoverflow 2015.).

3.7 Mobile device view

Surfing from a mobile device it is very common these days. Bootstrap responsive design was used in the application. Mobile screenshot is shown in Image 7. Using bootstrap components was used and very important in the application. For example, when image is not inside a bootstrap component, when trying to view the website from the mobile device, the image will show big and very uncomfortable to use.

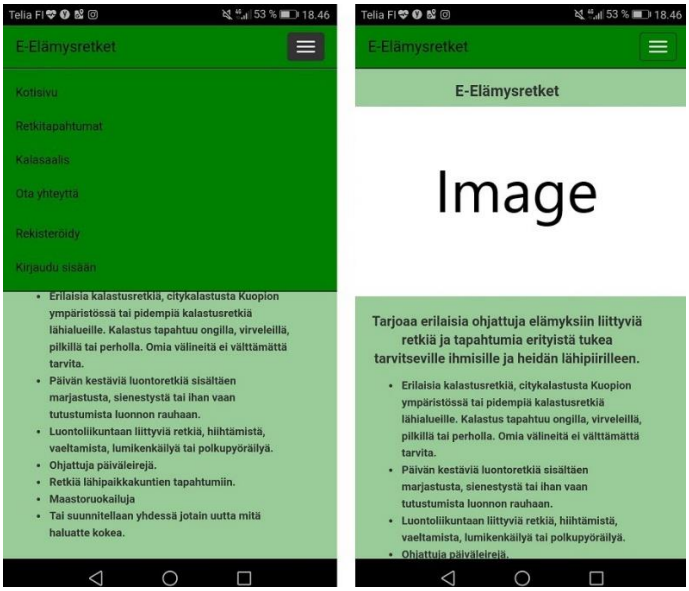

IMAGE 7. A mobile device view of the main page of the application.

4 PAGES

4.1 Activity page

The list on the activity-page, the blue text on the right side that is shown in Image 8, was organized from the newest to oldest added activity to the system.

Activity page display the following information: date when added, going on date, description, place name, city, pictures and comments and replies. Registration for going on activity is possible after login to the system (see Image 8). As well, it is possible to cancel the participation for the activity until the going on date expired.

IMAGE 8. The register to the activity button shows when the date is still on going.

4.2 Fish-page

The list on the Fish-page, the blue text on the right side that is shown in Image 9, was organized from the newest to oldest added fish to the system. Fish-page displayed the details of the chosen fish, such as the fisher, fish name, place, trap name, weight, length and the uploaded pictures.

Customers were able to be browsed the fish by clicking on the wanted fish from the fish list or with the page's numbers. As well, a logged in customer has the option to add his fish by clicking the button "Add own fish".

IMAGE 9. Fish-page screenshot with a logged in user.

4.3 Registration and login

4.3.1 Registration page

Registration to the system was required for the customer to be able to login. Customer had an option to choose to register by Facebook or to register with local account. Screenshots of the pages are shown in Image 10.

First name, last name, email and password are required for the registration. If customer miss one of them, application shows a specific detailed error. After registration with a local account, customer must to confirm his email before being able to login to the website.

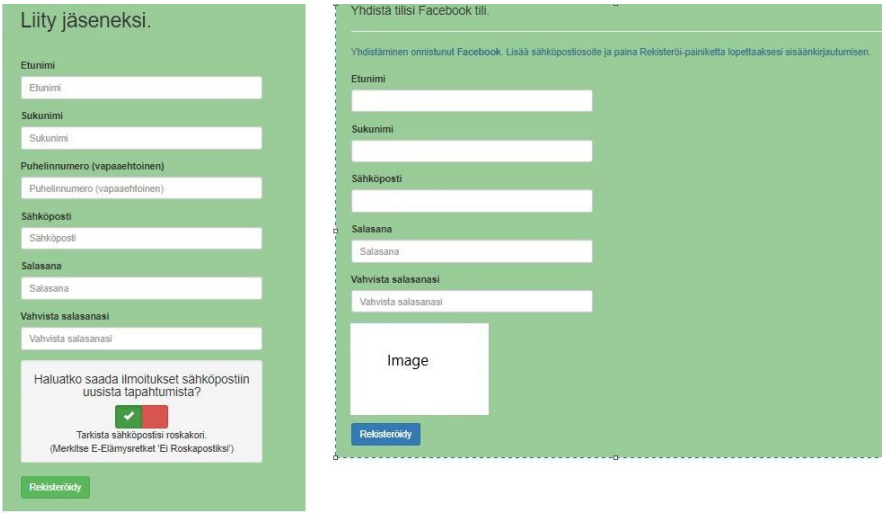

IMAGE 10. On the right side, the Facebook registration page. On the left registration page without Facebook.

4.3.2 Login page

In case of non-matched passwords or username or event if user was not found, login page was shown a proper error with a detailed message. Login page layout was taken from MIT license layout bootsnip.com page. Login page layout screenshot is shown in Image 11. (Bootsnipp 2012.)

Login page options:

- Email confirmation resent option,
- Sends to customer a new email with a new confirmation link code.
- Register to the system.

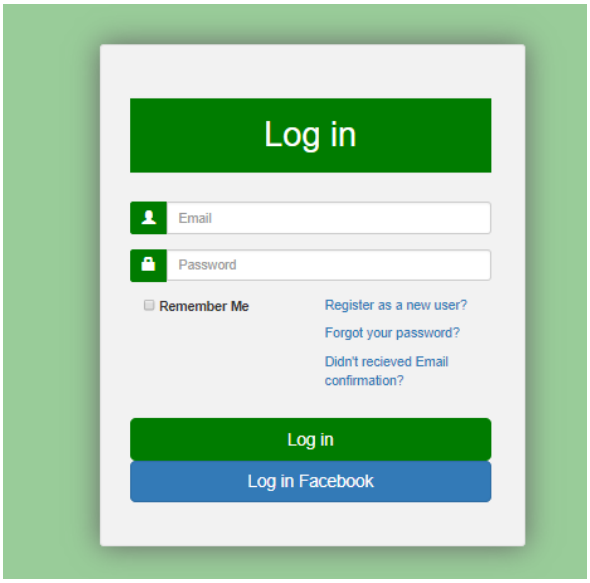

IMAGE 11. Login page screenshot.

4.4 Contact us

Customers were able to suggest an activity, ask a question or write a message to the company by filling the Contact Us page with the needed information. The code behind the contact us page takes all the customer's or guest's information that was added to the page and sends an email message to the company.

With this option, customers or guests can send a message to the company only by filling the page and after sending the message by Contact Us page, the system automatically sends a confirmation email to the customer or guest.

Contact Us page layout was modified from bootsnip.com page with a MIT license (Bootsnipp 2017). All the company's information is showed on the page for the customers. A page screenshot is shown in Image 12.

IMAGE 12. Contact-us page screenshot with logged in user.

4.5 Profile page

A logged in user can only view the own profile page. When user changes email information, the system checks if the email was not already given before and was not found in the database by another user.

Users can change the following information:

- First and last name.
- Email (sends a new email confirmation link email).
- Phone number.
- Password.
- Cancel or connect account with Facebook.
- User's profile picture.
- Check own participation in activities.

LINQ was used to get all logged in user's information. This way the code itself is short, but execute takes the same time as non- LINQ code. Code example of using LINQ is showin in Image 13.

```
ViewData["fishes"] = _context.FishDB.Where(fishi => fishi.UserID == user.Id).ToList();
```

IMAGE 13. Using Linq technic to get all current user's fishes and converted to list.

Customer view page screenshot is shown in Image 14. Extra options for customers only:

- Comments control, users can check if replied, agreed and delete each comment they have left.
- Fish control, customer can browse, edit and delete own added fish.

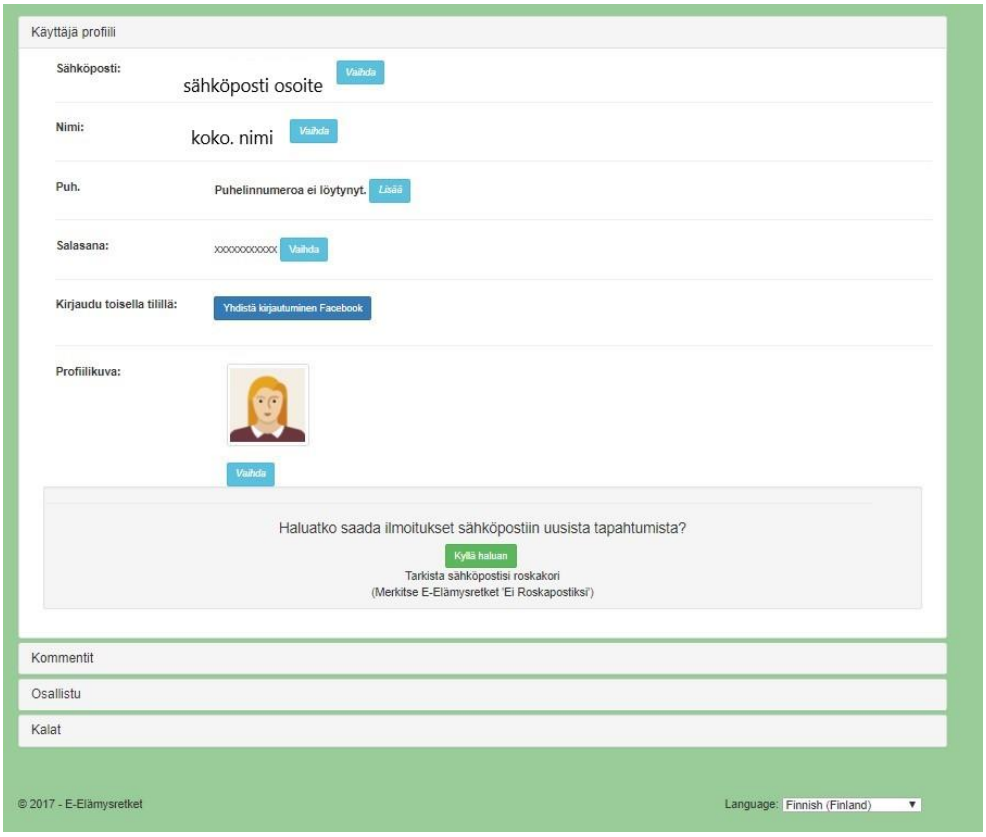

IMAGE 14. Customer profile page screenshot.

Admin user have an option to get all files with size information from specific folders in the root folder where all the application located. Getting all files information been used the IFileProvider extension. Code example is show in Image 15.

```
IFileProvider provider = new PhysicalFileProvider(_hostingEnvironment.WebRootPath);
IDirectoryContents userpics = provider.GetDirectoryContents("usersProfilePicFB");
```

IMAGE 15. Example of using IFileProvider, which get all files from "usersProfilePicFB" folder from the root folder.

Business information, Participants and registered emails for notifications works as well by taking in the current moment information from the database and convert them to a list parameter.

In the explanation bellow, you could see that parameter 'emails' type 'var' parameter. "var" means the compiler will determine the explicit type of the variable, based on usage. The parameter 'email' became a List type, that takes all the information from Notifications' table.

Code example is shown in Image 16. A paging feature is currently not implemented and might be added in the future.

```
var emails = _context.Notifications.ToList();
ViewData["emails"] = emails;

var participants = _context.Participants.ToList();
ViewData["participants"] = participants;
```

IMAGE 16. Example of getting information from the database and converting to a list type parameter.

Owner view page screenshot is shown in Image 17. Extra options for Admin only:

- Company information, admin can change company name, email, main page company's details text and main page images.
- Folders information, admin, can view specific folders size and their files.
- Notifications, admin can view user's who has been added to email notification system.

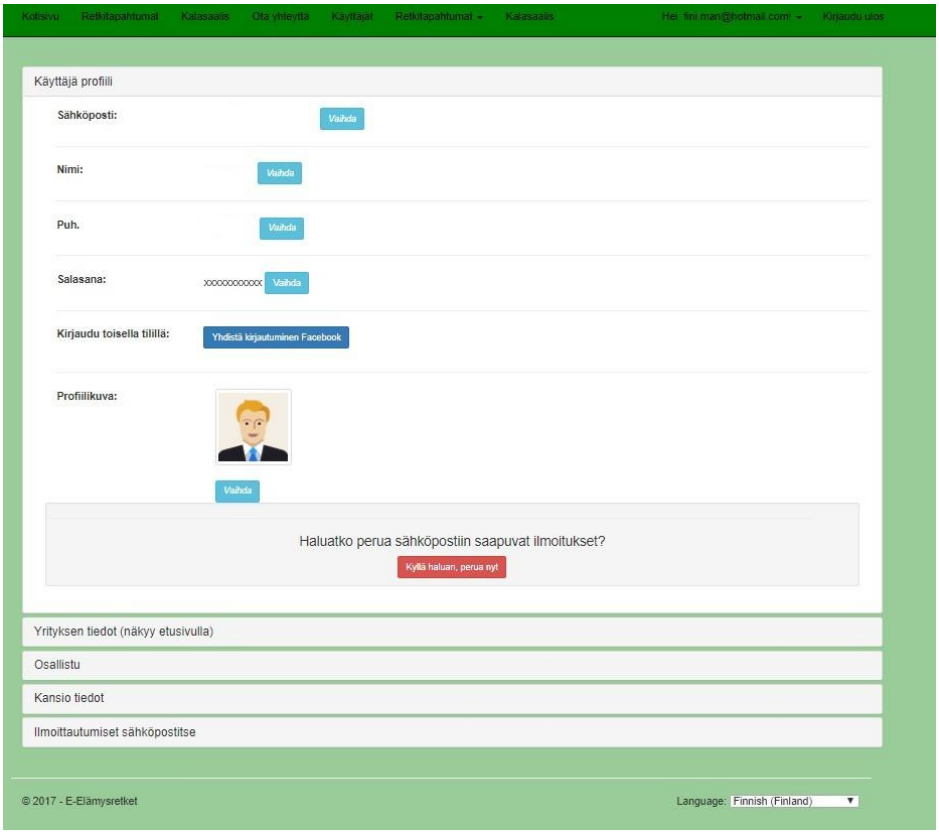

IMAGE 17. Admin own profile page.

4.6 Admin pages

4.6.1 Fish control panel page

The page screenshot is shown in Image 18. This page includes the following buttons and their functionality:

- Show images: show in a new small window all the images of the current fish.
- Edit: will move to a new page for with ready added information of the fish, and ready to be updated.

- Delete: will pop up a small question window, which will show a warning of deleting current fish. This warning message shows because, once the user deleted the current fish, all images and information will be deleted permanently.

IMAGE 18. Fishs control page.

4.6.2 Activities control page

In this page, the admin can edit, publish and unpublish, send a new activity email notification, edit the extra information and pictures and as well to delete each activity. If the admin decides to delete an activity, all pictures, comments and replies will be deleted as well from the system.

Page screenshot is shown in Image 19. This page includes the following buttons and their functionality:

- Edit activity: redirect to a new page with the inserted activity's information and ready to be edit.
- Show images: show in a small new window the images which have been uploaded to the system. Images will show only in 50% height and 30% width, to save space in the windows to show them all nicely and clearly.
- Show extra information of activity: shows in a small window the inserted extra information.
- Publish and unpublish: publishing means, the current activity will be shown in the activities main page. Same works for the unpublishing button, which will take the current activity out from the activities main page.
  After creating a new activity, the default is always unpublish. Which gives the admin an option to go thru the inserted details and confirm them.
- Send emails notification: shows a draft of the email which is going to be send to all registered emails in the database, admin will have to give a title for the email which will be shown as the email subject for customers.
- Edit or Adding extra information: redirect to a new page, where user can add or edit the extra information.
- Delete: show a warning message to user with information of all data will be deleted as well.

IMAGE 19. Activities control page.

4.6.3 Replies control page

Page screenshot is shown in Image 20. This page includes the following buttons and their functionality:

- Edit: will forward to a new page where the comment text can be.
- Delete: will delete the current comment along with the reply if found.

IMAGE 20. Replies control page.

4.6.4 Comments control page:

Page screenshot is shown in Image 21. This page includes the following buttons:

- Agree and Unagreed: After customer comment on activity, admin can go first thru the comment, and agreed it. Which will make the comment visible in the current activity admin page.
- Replied and not reply: user can mark the comment as a replied, which means the comment don't need any reply. Then in the main page, admin user will see that there is no need to reply the current comment. As well, the comment will be marked as replied for admin and customer users.
- Spam unspam: by clicking on this button, the comment will be marked as a spam which means the comment will not show in the activity main page. For releasing the comment from a spam option, there is the unspam button.
- Delete: show a warning message to make sure user want to delete the comment. While deleting the comment, if found reply for this comment, it will be deleted as well.

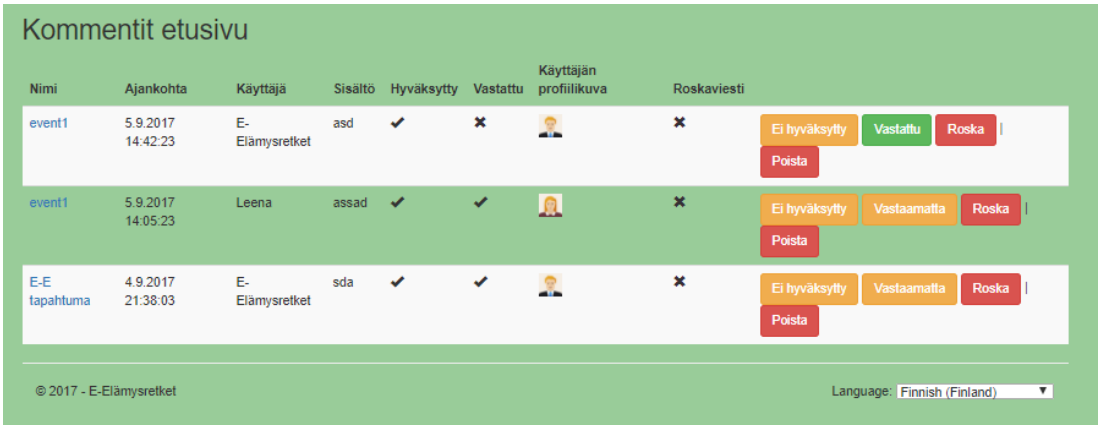

IMAGE 21. Comments control page.

4.6.5 Activities control panel

In this control panel, shows new comments, unagreed comments, newest customer comment, published and unpublished activities, replies, participants amount and last 10 participants and their activities.

Activities control panel layout has been taken from sb-admin 2 template, which has MIT licence. (Github 2015). Every new information was added to the database, this control panel will be updated as well. Page screenshot is shown in Image 22.

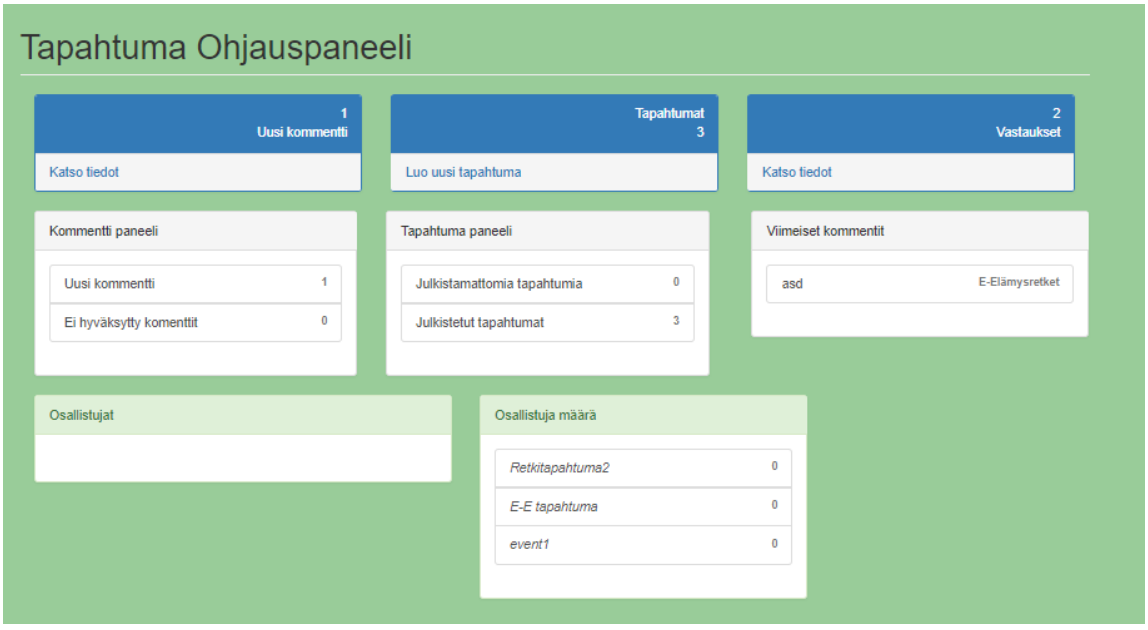

IMAGE 22. Activities control panel screenshot.

4.6.6 Users control page

Page screenshot is shown in Image 23. Page includes the following buttons:

- Reset password: resetting password, will reset the password for wanted user with a specific password, which admin will get a notification of the password information.
- Delete: by clicking the delete button, all comments, user's uploaded fishes and users account information will be totally deleted.

IMAGE 23. Users control list screenshot.

5 HOSTING AND DOMAIN

Looking for the right hosting company was hard. There are many companies on the internet, but most of them didn't have enough infomarion to verify their reliability. The searching for the right hosting company was narrowed down to two companies , Azure of Microsoft and Smarterasp.net. Azure's website was of very high quality but complicated to understand how features are used and it was more expensive. Smarterasp.net have a very simple guide to publish asp.net core applications to the server. (Smarterasp.net 2016.)

Support for Asp.net Core 2.0 and MSSQL database were the hosing requirements for this application. Many companies were not being able to support these features.

Finding a proper hosting company was the last part of the thesis. The hosting company for this application was Smarterasp.net, which provided a very good customer service and support when needed with cheap prices. (Smarterasp.net 2017.)

Domain and the Secure Sockets Layer (SSL) was added as well from the Smarterasp.net site, the company was taken care of combining the new domain to their system.

TABLE 1. Comparison between hosting companies. All prices were taken at 20 September 2017.

Hosting company/ features	Azure (Microsoft)	Smarterasp.net	Bluehost	GoDaddy
Support asp.net core 2	Yes	Yes	No	No
Database type support	MSSQL and MySQL	MSSQL and MySQL	MySQL	MySQL
Database size	2GB	1GB	1GB	1GB
Includes domain	No	No	Yes	Yes
Includes custom email account	No	Yes	No	No
Price (monthly)	Start from DB: 4.21€ Site hosting: 8.16€	Start from 2.30€	Start from 2.30€	Start from 2.49€

6 FURTHER DEVELOPMENT

More features and new options can be created and added to this application. All of the features that was required by the E-Elämysretket company are done and working.

Application can be supported with more languages. Login with Google+, Instagram, Twitter and LinkedIn can be added in the future and there is no need to add any new classes or views.

OneDrive and Google drive can be added to the application, to make uploading and choosing images more effective for the company and customer.

Paging, which means information sorting, can be added as well in the future, when the company will get bigger and will have more customers. The application can be controlled easier and in a safer way while getting the information from the database by adding the paging option. This application does not support paging, but still works as planned.

7 SUMMARY

By coding this application, a lot of new information was studied, which will be very useful for future developing. At first, the application plan was very well described. During working on the application, E-Elämysretket manager asked me to add and fix small things, such as colours of the website, lay-out and possibility to login and register with the Facebook account.

The application's database was modified many times to get the most effective database for the company's needs. The database in the application was built with ASP.NET Core 2.0 framework, basically the framework created automatically the users and roles tables. These tables were modified during the developing.

The application was included two kinds of main layouts, one for the user view and another one for the admin user only. These two layouts made the website simple and safe to use. Inside each of these layouts was a language change drop list selection option, which switch to the wanted language. This application was able to recognize (mostly in chrome and Edge explorers) what language users used in the device.

All necessary features were done successfully, such us the Facebook login and registration, controlling the database information and switching between two languages. E-Elämysretket acknowledged the site and the hosting company.

REFERENCES

BOOTSTRAP. 2017. Bootstrap. [reference made: 2017-10-04]. Available at: http://getbootstrap.com/.

BOOTSNIPP. 2012. Login with show/hide password feature. [reference made: 2017-08-05]. Available at: https://bootsnipp.com/snippets/v24ze.

BOOTSNIPP. 2017. Contact us page. [reference made: 2017-07-08]. Available at: https://bootsnipp.com/snippets/4lw12.

CORECOMPAT. 2017. CoreCompat/System.Drawing. [reference made: 2017-06-04]. Available at: https://github.com/corecompat/system.drawing/blob/master/license.txt.

FACEBOOK. 2017. Apps. [reference made: 01.08.2017]. Available at: https://developers.facebook.com/apps/.

FRAMEWORK. 2013. Framework Definition. [reference made: 2017-08-20]. Available at: https://techterms.com/definition/framework.

GITHUB. 2015. Mailkit. [reference made: 2017-06-20]. Available at: https://github.com/jstedfast/mailkit.

GITHUB. 2015. Sb-admin 2 bootstrap template for asp mvc. [reference made: 2017-06-20]. Available at: https://github.com/lvasquez/sb-admin-2-bootstrap-template-asp-mvc.

MAILKIT. 2017. Mailkit. [reference made: 2017-06-06]. Available at: https://www.nuget.org/packages/mailkit/.

MICROSOFT. 2017. Announcing asp net core 2 0. [reference made: 2017-06-01]. Available at: https://blogs.msdn.microsoft.com/webdev/2017/08/14/announcing-asp-net-core-2-0/.

MICROSOFT. 2017. Localization. [reference made: 2017-06-18]. Available at: https://docs.microsoft.com/en-us/aspnet/core/fundamentals/localization.

MICROSOFT. 2017. System.io.file. [reference made: 2017-07-01]. Available at: https://msdn.microsoft.com/en-us/library/system.io.file(v=vs.110).aspx.

MICROSOFT. 2017. Entity Framework Core Quick Overview. [reference made: 2017-09-21]. Available at: https://docs.microsoft.com/en-us/ef/core/.

MICROSOFT. 2006-03-29. What is a Database. [reference made: 2017-10-04]. Available at: https://channel9.msdn.com/Blogs/ASP-NET-Site-Videos/what-is-a-database.

MICROSOFT. 2007-02. LINQ: .NET Language-Integrated Query. [reference made: 2017-09-19]. Available at: https://msdn.microsoft.com/en-us/library/bb308959.aspx.

MICROSOFT. 2003. C# and Java. [reference made: 2017-10-04]. Available at: https://msdn.microsoft.com/en-us/library/ms836794.aspx.

MICROSOFT. 2008-08-19. Understanding Models, Views, and Controllers (C#). [reference made: 2017-10-04]. Available at: https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/overview/understanding-models-views-and-controllers-cs.

SMARTERASP.NET. 2016. How to use web deploy feature via visual studio 2015. [reference made: 2017-08-20]. Available at: http://www.smarterasp.net/support/kb/a1773/how-to-use-web-deploy-feature-via-visual-studio-2015.aspx?kbsearchid=265106.

SMARTERASP.NET. 2017. [reference made: 2017-08-15]. Available at: http://www.smarterasp.net/.

STACKOVERFLOW.COM. 2015. How to send email by using MailKit. [reference made: 2017-06-20]. Available at: https://stackoverflow.com/questions/33496290/how-to-send-email-by-using-mailkit.

VISUAL STUDIO MICROSOFT. 2017. Visual studio 2017 release notes. [reference made: 2017-06-01]. Available at: https://www.visualstudio.com/en-us/news/releasenotes/vs2017-relnotes.