March 2016

Анализ на общинското присъствие във Facebook

**Contents**

[Увод 3](#_Toc461383983)

[Related work 5](#_Toc461383984)

[Detailed requirements 5](#_Toc461383985)

[Functional requirements 6](#_Toc461383986)

[User authentication 6](#_Toc461383987)

[Capture and present general data about the municipality 6](#_Toc461383988)

[Convenient data browsing 6](#_Toc461383989)

[Open data 6](#_Toc461383990)

[Non-functional requirements 8](#_Toc461383991)

[Database management - easy addition of columns 8](#_Toc461383992)

[Automated regression tests 8](#_Toc461383993)

[Application deployment 8](#_Toc461383994)

[Architecture 9](#_Toc461383995)

[Data access layer 9](#_Toc461383996)

[.NET entity framework 9](#_Toc461383997)

[Web services 12](#_Toc461383998)

[.NET web services used for AJAX calls/requests 12](#_Toc461383999)

[Presentation layer 18](#_Toc461384000)

[ASP.NET 18](#_Toc461384001)

[ASP.NET Dynamic Data 19](#_Toc461384002)

[Data Layer 19](#_Toc461384003)

[Server – MS SQL server 19](#_Toc461384004)

[Schema – normalized DB with synthetic keys 19](#_Toc461384005)

[Objects and their relations 19](#_Toc461384006)

[References 20](#_Toc461384007)

# Увод

Фейсбук (на английски Facebook) е най-голямата онлайн социална мрежа в света, която поддържа повече от 70 езика. Достъпен от 2004 г. насам, днес той има над 223 милиона потребители в Европа и постави рекорд в края на месец август тази година с над 1 милиард активни потребители за ден. Повече от 70% от потребителите на тази социална мрежа констатират, че я посещават поне веднъж дневно като над 45% от тях влизат във Фейсбук по няколко пъти на ден. Това е една от социалните мрежи с най-лесен и достъпен начин за представяне на информация и комуникация (Уикипедия, 2015).

Това я нарежда на първо място сред социалните мрежи и е съвсем логично желанието за присъствие на държавните институции в Фейсбук и в частност на общините, както в България, така и на международно ниво. Мрежата предоставя всички необходими средства за представянето, информирането и обратната връзка на общините със заинтересованите Фейсбук потребители. Въпреки, че това не е държавна политика, някои от българските общини имат присъствие във Фейсбук под една или друга форма. Не винаги обаче то е на нивото, което позволява лесно намиране на страницата на общината, информацията за общината и проектите или инициативите, които тя организира. Обратната връзка със заинтересуваните потребители и диалогът, който социалната мрежа позволява да бъде проведен с тях, са средства, които биват подценявани и не достатъчно развити в присъстващите в Фейсбук общински страници.

Целта на дипломната работа еанализ на присъствието на общините във Фейсбук и на това как те могат да станат по-популярни в тази социална мрежа, как да извлекат максималната полза от представянето си, къде се намират в сравнение с други общини по света и какво още могат да направят за подобряване на връзката община – население.

Анализът е базиран на важни характеристики, предоставени от Фейсбук като ще бъде подробно изследвана всяка от тях и какво е нейното значение за успех сред потребителите. На базата на тези характеристики, ще бъде разгледано присъствието на българските общините във Фейсбук към този момент, сравнено с минали периоди и съпоставено с общини на други държави по света.

Различни анализи, сравнения и заключения на базата на въведени данни, ще се реализират чрез единна система, предоставяща възможност на всеки да попълва информация за дадена община на базата на представянето й във Фейсбук. Заключенията от анализите направени от тази система, ще дадат възможност да се подобри процесът по търсене, представяне и взаимодействие между общините и гражданите – потребители на Фейсбук.

Задачи, произтичащи от целта:

* Създаване на система, която ще предоставя възможност на всеки потребител да въвежда предварително дефиниран формат на информация за намерена община във Фейсбук, ще има възможност да прави търсения на база въведената информация за общините до момента и да сравнява показателите на всяка една от тях.
* Създаване на интерфейс за наблюдение на тенденциите на всяка община и за сравняване и анализ на резултатите.
* Системата ще има административна част, в която ще се даде възможност информацията попълнена за дадена община да бъде проверена преди тя да бъде използвана за целите на търсене и анализ.
* Системата ще има база от данни, в която ще се съдържа въведената и проверена за вярност информация, както и ще съдържа методи, по които да се проверява дали информацията е в правилния формат.

Facebook is the biggest online social network which supports more than 70 languages. Launched in 2004, today there are more than 223 million of people in Europe using it (Zephoria Digital Marketing, 2015). In August last year Facebook hits a new peak of 1 billion active users for a day (Matney, 2015). More than 70% of Facebook users state that they log in the social network at least once a day and over 45% of them are using Facebook several times per day. This is one of the social networks with the most user friendly interface for information presence and communication

These advantages make Facebook the most desirable network for municipalities all over the world as place for their social presence and communication with interested in their work Facebook users.

Municipalities’ presence in this social network gives an opportunity for analyses over their current presentation, comparison between municipalities all over the world, how their Facebook presence can become more profitable, their Facebook sites become more popular and their Facebook followers become more active.

A web application which will help analyzing the data for different municipalities will be created. Users will be able to log in the application with their Facebook accounts and enter information for municipalities’ pages in Facebook. The data then will be reviewed by administrators and the information will be used for comparison between different municipalities all over the word. There will be an interface for review of the trends of these Facebook pages, sorting of the data and methods for free reuse of the gathered data.

# Related work

# Detailed requirements

Requirements are separated to two different types – functional and non-functional. Functional requirements are the use cases and the application implements. Non-functional requirements enable the continuous maintenance, enhancement and operation of the application.

## Functional requirements

### User authentication

All users need to be authenticated based on their email or social network membership. Currently only a Facebook authentication is implemented. Administrators are separated by regular users using a flag in their data. Only existing administrators can make other users administrators of the application.

### Capture and present general data about the municipality

Before the social presence of the municipality can be entered contributors enter general information like name, country, population and website. All visitors can see data for all municipalities. They can also see whether the data is approved by the administrator. Only administrators can approve or edit the data of an already recorded municipality.

### Convenient data browsing

User should be able to sort all data lists by any attribute. Both forward and reversed sorting should be supported. Text attributes need to be sorted alphabetically while numeric attributes by magnitude. Each attribute name should have a tooltip that explains in more details its meaning.

### Open data

Visitors can export the data in a standard and an open format. The csv format was selected because of its wide support. Pages from the application can be dynamically embedded in other portals.

1. Municipality column - first in the grid view home page – On the municipality page
2. Edit button to be hidden – Only on Page load
3. Details for the user to be added – name, email, phone number
4. Email verification, send email to user with filled in data
5. Column sorting
6. Changing columns functionality – user can reorder the columns so it is more comfortable to see the data – nice to have it
7. Optional: Information for all countries and their municipalities
8. Database requirement: easy addition of columns
9. Tooltips with explanations for the fill in form(on user insert)
10. Validations: emails, hyperlinks…
11. New DB table: group information on years: date of inserting data in the application.
12. Abandoned sites: not active for a long time : display such sites
13. Generate information in excel file(open data)
14. Import to your site functionality
15. Charts- the same as in the excel spreadsheet
16. Change date column with calendar control
17. Encoding in the excel file
18. Admin functionality ( DB changes)
19. Scroll back – edit information
20. DB Model
21. Solution Architecture + text in English
22. Description of functionality(more details for solution) in English
23. User manual
24. Columns in the datagrid to be displayed with user friendly names not the ones form the DB(Nice to have)
25. Authomated import of data for further improvements
26. Query to DB views (another import option – open data)
27. Line at the bottom of the page for the license of the data(open data)

## Non-functional requirements

### Database management - easy addition of columns

Using Entity framework and ASP.NET Dynamic data allows automate adding of new columns to the object model and the user interface. A system administrator can add a column to the MS SQL database and then expose it in the UI without any code changes by just synchronizing the object mode with the database.

### Automated regression tests

To enable future maintenance of the application, it is essential to allow developers to make changes in one part of the application with reduced risk of making regression in another part. To achieve that the application is shipped with a suite automated unit and integration tests.

### Application deployment

The application needs to be packaged in a way that enables save and automated deployment to any server that supports ASP.NET.

# Architecture



Architecture of our project is based on three layers: Data access layer, Presentation layer and Data layer. Below we will review all of them in more details.

## Data access layer

### .NET entity framework

In order to make our application flexible and easy to maintain we use ORM.

What is O/RM?

ORM is a tool for storing data from domain objects to relational database like MS SQL Server, in an automated way, without much programming. O/RM includes three main parts: Domain class objects, Relational database objects and Mapping information on how domain objects map to relational database objects (tables, views & stored procedures). ORM allows us to keep our database design separate from our domain class design. This makes the application maintainable and extendable. It also automates standard CRUD operation (Create, Read, Update & Delete) so that the developer doesn't need to write it manually.

A typical ORM tool generates classes for the database interaction for our application as shown below.

[](http://www.entityframeworktutorial.net/Images/ORM.png)

There are many ORM frameworks for .net in the market such as DataObjects.Net, NHibernate, OpenAccess, SubSonic etc. Entity Framework is an open source ORM framework from Microsoft and we use it in our project. (Entity Framework Tutorial, 2016)

What is Entity Framework?

Writing and managing ADO.Net code for data access is a tedious and monotonous job. Microsoft has provided an O/RM framework called "Entity Framework" to automate database related activities for your application.

Microsoft has given the following definition of Entity Framework:

The Microsoft ADO.NET Entity Framework is an Object/Relational Mapping (ORM) framework that enables developers to work with relational data as domain-specific objects, eliminating the need for most of the data access plumbing code that developers usually need to write. Using the Entity Framework, developers issue queries using LINQ, then retrieve and manipulate data as strongly typed objects. The Entity Framework's ORM implementation provides services like change tracking, identity resolution, lazy loading, and query translation so that developers can focus on their application-specific business logic rather than the data access fundamentals.

Entity framework is an Object/Relational Mapping (O/RM) framework. It is an enhancement to ADO.NET that gives developers an automated mechanism for accessing & storing the data in the database.

Entity framework is useful in three scenarios. First, if you already have existing database or you want to design your database ahead of other parts of the application. Second, you want to focus on your domain classes and then create the database from your domain classes. Third, you want to design your database schema on the visual designer and then create the database and classes.

The following figure illustrates the above scenarios.

[](http://www.entityframeworktutorial.net/Images/EF-overview.png)

As per the above figure, EF creates data access classes for your existing database, so that you can use these classes to interact with the database instead of ADO.Net directly.

EF can also create the database from your domain classes, thus you can focus on your domain-driven design.

EF provides you a model designer where you can design your DB model and then EF creates database and classes based on your DB model. (Entity Framework Tutorial, 2016)

In our project we used the first scenario. At first the database was created and then Data access classes were generated through Entity Framework.

## Web services

### .NET web services used for AJAX calls/requests

Add Facebook Login to Your App or Website

Facebook Login for Apps is a secure, fast and convenient way for people to log into your app or website. (Facebook, 2016)

Facebook Login for Apps is a fast and convenient way for people to create accounts and log into your app across multiple platforms. It's available on [iOS](https://developers.facebook.com/docs/ios/login/), [Android](https://developers.facebook.com/docs/android/login-with-facebook), [Web](https://developers.facebook.com/docs/facebook-login/web), [Windows Phone](https://developers.facebook.com/docs/facebook-login/windows-phone), [desktop apps](https://developers.facebook.com/docs/facebook-login/manually-build-a-login-flow) and [devices such as Smart TVs and internet-of-things objects](https://developers.facebook.com/docs/facebook-login/for-devices).



Use Cases

Facebook Login is used to enable the following experiences:

1. **Account Creation**  
   Facebook Login lets people quickly and easily create an account in your app without having to set (and likely later forget) a password. This simple and convenient experience leads to higher conversion. Once someone has created an account on one platform, they can log into your app - often with a single click - on all your other platforms. A validated email address means you're able to reach that person to re-engage them at a later date.
2. **Personalization**  
   Personalized experiences are more engaging and lead to higher retention. Facebook Login lets you access information which would be complex or arduous to collect via your own registration form, for example: the things a person has liked, their birthday, their hometown or current location, their work history. Even just importing someone's profile picture imported from Facebook gives them a stronger sense of connection with your app.
3. **Social**  
   Many highly retentive apps let people connect with their friends in order to enable shared in-app experiences. Facebook Login lets you know which of your app's users are also friends on Facebook so you can build value by connecting people together.

Features

1. **Real Identity**  
   When people choose to log in with Facebook, they can share their real identity through their public profile. Public Profile includes a person's real name, a profile picture, their gender and their locale. Apps based on real identity often have less spam and foster higher quality conversations.
2. **Cross Platform Login**  
   Facebook Login is available on the most common mobile and desktop app platforms. People who create accounts with Facebook on one platform can quickly and easily log into your app on another. A person is known by the same User ID everywhere so they can pickup your in-app experience where they left off. Facebook Login is available on [iOS](https://developers.facebook.com/docs/ios/login/), [Android](https://developers.facebook.com/docs/android/login-with-facebook), [Web](https://developers.facebook.com/docs/facebook-login/web), and [Windows Phone](https://developers.facebook.com/docs/facebook-login/windows-phone), for [desktop apps](https://developers.facebook.com/docs/facebook-login/manually-build-a-login-flow) and for [devices such as Smart TVs and internet-of-things objects](https://developers.facebook.com/docs/facebook-login/for-devices)
3. **Works Alongside Your Existing Account System**  
   Facebook Login complements your existing account system. Give people the option to log in with Facebook alongside email, SMS or other social login choices. Where an email address you get from Facebook Login matches one already in your system, you can log that person into their existing account without additional passwords.
4. **Granular Permissions**  
   Facebook Login supports [over 30 permissions](https://developers.facebook.com/docs/facebook-login/permissions) which determine which information people share with your app. This means you have precice control over what you request, and what people choose to approve.
5. **People Have Control over What They Share**  
   Great experiences start by giving people control. With Facebook Login, people can choose which information they share with your app. They can still get the benefits of logging in with Facebook even if they feel uncomfortable granting access to certain information. Your app can later re-request this information once you've explained how the person's experience will be enhanced.
6. **Gradual Authorization**  
   Facebook Login supports the gradual authorization - you don't have to request all the information you want up front - you can do it over time. This means people can quickly and easily create accounts in your app - and as their experience with your app deepens, you can request addition information to further enhance their experience.

Login Review

We want to ensure the tens of millions of people who use Facebook Login every day have a safe, reliable and consistent experience. Login Review helps determine if apps that are requesting access to detailed account information have built great experiences.

Our Login Review process is designed to be fast and lightweight. Our review team actually use your app and will provide guidance and feedback to ensure you're meeting our [Platform Policies](https://developers.facebook.com/policy).

Apps may ask for the following three permissions from any person **without** submitting for review by Facebook:

* [public profile](https://developers.facebook.com/docs/facebook-login/permissions#reference-public-profile)
* [email](https://developers.facebook.com/docs/facebook-login/permissions#reference-email)
* [user\_friends](https://developers.facebook.com/docs/facebook-login/permissions#reference-friends)

To ask for any other permission, your app will need to be [reviewed by Facebook](https://developers.facebook.com/docs/facebook-login/review) before these permission become visible in the Login Dialog to the public who're logging into your app with Facebook.

However, to help you develop your Facebook Login experience, anyone listed in the Roles section of your app's dashboard will be able to grant any valid [permission](https://developers.facebook.com/docs/facebook-login/permissions) without approval from Facebook.

Learn more about [Login Review](https://developers.facebook.com/docs/facebook-login/review). (Facebook, 2016)

Definition - What does Web Service mean?

A Web service, in the context of .NET, is a component that resides on a Web server and provides information and services to other network applications using standard Web protocols such as HTTP and Simple Object Access Protocol (SOAP).  
  
.NET Web services provide asynchronous communications for XML applications that operate over a .NET communications framework. They exist so that users on the Internet can use applications that are not dependent on their local operating system or hardware and are generally browser-based.

Techopedia explains Web Service

The main advantage of a Web service is that its consumers can use the service without knowing about the details of its implementation, such as the hardware platform, programming language, object model, etc. Web service provides a loose coupling between heterogeneous systems with the help of XML messages, provide interoperability.  
  
Web services are designed to provide the messaging infrastructure necessary for communication across platforms using industry standards. Web services also use asynchronous communication to address the latency issue that arises due to requests from remote locations across the Internet. This allows the execution of background tasks for the client (such as responding to user interactions) until the actual completion of the Web service request.  
  
ASP.NET provides a framework that can be used to build Web services easily by focusing on the application logic rather than on the hardware necessary for writing infrastructure code for communication protocol or message transport. Web services created in ASP.NET can use features of the .NET framework such as caching, authentication and state management.  
  
Web service uses the ".asmx" extension with the @Web service directive (at the top of file) as per the ASP.NET application model. It can be a stand-alone application or a subcomponent of a larger Web application. (Techopedia Inc., 2016)

 ( Stefan Bergstein, HP Software, 2016)

## Presentation layer

### ASP.NET

ASP.NET is a unified web development model integrated with .NET framework, designed to provide services to create dynamic web applications and web services. It is built on the Common Language Runtime (CLR) of the .NET framework and includes those benefits like multi-language interoperability, type safety, garbage collection and inheritance.   
  
Mark Anders and Scott Guthrie of Microsoft created the first version of ASP.NET in 1992. It was created to facilitate the development of distributed applications in structured and object-oriented manner by separating the presentation and content and hence write clean code. ASP.NET uses the code-behind model to generate dynamic pages based on Model-View-Controller architecture.   
They have some the major differences from ASP, an earlier version of ASP.NET. The object model of ASP.NET has thus significantly improved from ASP, which makes it fully backward compatible to ASP.   
  
These differences include:  
1. Usage of compiled code (instead of interpreted code),   
2. Event-driven server-side scripting model,   
3. State management,   
4. Rapid application development using controls and libraries of the .NET framework.   
5. Dynamic programming code is placed separately in a file or specially designated tag. This avoids the program code getting modified during runtime.

ASP.NET works with the Internet Information Server (IIS) to deliver the content in response to client requests. While processing the requests, ASP.NET provides access to all .NET classes, custom components and databases, similar to that of a desktop application.  
  
Web forms are the building blocks of application development in ASP.NET. They provide lot of flexibility by allowing controls to be used on a page as objects. These controls can handle events such as Load, Click and Change, similar to those in desktop applications. Other than Web forms, ASP.NET can be used to create XML Web services that can allow building modular, distributed web applications, written in any language.These services are interoperable across variety of platforms and devices.  
  
In addtion, ASP.NET implements state management by sending the information (viewstate) related to state of controls on a web form to the server in a postback request. It provides side-by-side execution applications of multiple denominations allowing them to be installed on the same system with different versions of .NET frameworks. Furthermore, it uses XML support for data storage, configuration and manipulation. However, when it comes to securing its applications, ASP.NET uses the code access security and role based security features of .NET framework and inherent methods of IIS for authenticating user credentials. (Techopedia Inc., 2016)

### ASP.NET Dynamic Data

## Data Layer

### Server – MS SQL server

### Schema – normalized DB with synthetic keys

### Objects and their relations

# References

Stefan Bergstein, HP Software. (2016, March 22). *Service Oriented Architectures (SOA) Monitoring and Management with HP OpenView Operations* . Retrieved from Slideshare: http://image.slidesharecdn.com/univ05ovosoaext-150828145753-lva1-app6891/95/service-oriented-architectures-soa-monitoring-and-management-with-hp-openview-operations-14-638.jpg?cb=1440773962

Entity Framework Tutorial. (2016, March 22). *What is Entity Framework?* Retrieved from Entity Framework Tutorial: http://www.entityframeworktutorial.net/what-is-entityframework.aspx

Facebook. (2016, March 22). *Add Facebook Login to Your App or Website* . Retrieved from Facwbook for developers: https://developers.facebook.com/docs/facebook-login

Facebook. (2016, March 22). *Facebook Login for Apps - Overview*. Retrieved from Facebook for developers: https://developers.facebook.com/docs/facebook-login/overview

Matney, L. (2015, November). *Facebook Hits New Peak Of 1 Billion Users On A Single Day*. Retrieved from http://techcrunch.com/2015/08/27/facebook-hits-1-billion-users-in-a-single-day/#.mkj389:Hrf0

Techopedia Inc. (2016, March 22). *Web Service*. Retrieved from Techopedia: https://www.techopedia.com/definition/1622/web-service-net

Zephoria Digital Marketing. (2015, November). *The Top 20 Valuable Facebook Statistics*. Retrieved from https://zephoria.com/top-15-valuable-facebook-statistics/

Уикипедия. (2015, November). *Фейсбук*. Retrieved from https://bg.wikipedia.org/wiki/%D0%A4%D0%B5%D0%B9%D1%81%D0%B1%D1%83%D0%BA