Babeș-Bolyai University Cluj-Napoca

Faculty of Mathematics and Computer Science

Specialization Computer Science English

Diploma Thesis

**Nume aici**

**Supervisor**

Conf. Univ. Dr. Niculescu Virginia

**Author**

Laslo Marian

Cluj Napoca

**2017**

Universitatea Babeș-Bolyai Cluj-Napoca

Facultatea de Matematică și Informatică

Specializare Informatică Engleză

Lucrare de Licență

**Nume aici**

**Supervisor**

Conf. Univ. Dr. Niculescu Virginia

**Author**

Laslo Marian

Cluj Napoca

**2017**

Contents

[1. Introduction 4](#_Toc485254529)

[2. Web Applications 5](#_Toc485254530)

[2.1 What is a Web Application? 5](#_Toc485254531)

[2.2 Design Patterns 5](#_Toc485254532)

[2.3 The Model-View-Controller pattern 5](#_Toc485254533)

[3. Web Applications using .NET 6](#_Toc485254534)

[3.1 What is .NET 6](#_Toc485254535)

[3.2 Advantages of using .NET in a web application 6](#_Toc485254536)

[3.3 Most common .NET Frameworks for developing web applications 7](#_Toc485254537)

[3.4 ASP.NET MVC versus Web-API 7](#_Toc485254538)

[3.5 Conclusion 8](#_Toc485254539)

[4. Application 8](#_Toc485254540)

[4.1 Development Process 8](#_Toc485254541)

[4.2 The final result 10](#_Toc485254542)

[Bibliography 11](#_Toc485254543)

# Introduction

The purpose of this thesis is help an IT company look for potential interns. The application should make the process of recruiting interns a lot easier for the company and for the members assigned to recruit the interns.

**Why a web application?**

Compared to the standard desktop application, which has to be installed on a computer in order to work, the web application only requires an internet connection and access to the website that hosts the application in order to be used.

A good example of an application that has both desktop and web support is Slack. In order to use Slack, you can either install the application on your desktop or, you can use the web application for Slack and just access the website, login and start using it from the browser. The main advantage is that you don’t have to install anything else on the computer in order to use the application.

Another advantage of a web application is that it can be used on a large variety of devices. Nowadays, when smartphones and mobile devices in general are so common, it is worth mentioning that a web application can be accessed from any device that has a browser, this includes all the smartphones and tablets available on the market.

Furthermore, a web application can be used from multiple devices that use different architecture without having to develop the application for each one of them, which is the case for desktop applications. A good example would be to create a desktop application for both Mac OS and Windows; you would basically have to develop the application for both operating systems separately.

This thesis contains the following chapters:

**Chapter 1** – this chapter focuses on presenting the purpose and the main objective of this thesis and it also presents the advantages of Web applications.

**Chapter 2** – presents some general information about Web applications and about the MVC framework from .NET which is used in the application.

**Chapter 3** – contains information about Web applications using .NET. It also presents and compares the frameworks available in .NET for Web applications.

**Chapter 4** – this chapter presents how the application itself is structured and the interaction between the user and the application

**Chapter 5** – presents some aspects of the application that can be improved

**Chapter 6** – conclusions regarding the application, presenting some of the more difficult parts of developing the application and some final opinions.

# Web Applications

## 2.1 What is a Web Application?

A web application is an application that uses a web browser as a client [1]. The application’s complexity can vary from a simple merchandise selling application, to a complex document manager. Web applications are written in browser-supported languages like HTML, JavaScript and TypeScript because they depend on the browser to execute them [2].

There are two types of web applications, static and dynamic [2]. For example, a presentation page which only contains information about something is usually a static application because there is no need for server-side processing. On the other hand, even if the web application only contains a login or register functionality, it’s going to be a dynamic application because it requires server-side processing.

## 2.2 Design Patterns

Some of the more commonly used design patterns in web applications (and not only) are MVC or Model-View-Controller, Singleton pattern, Adapter pattern, Inversion of Control, Dependency Injection and Façade pattern.

## 2.3 The Model-View-Controller pattern

What is it? MVC is a software architecture which separates the business logic of the application from the user interface. The application is separated into three parts: model, view and controller. A model represents knowledge and it could be a single object, or it could be a structure of multiple objects; it usually is a database of some kind [5]. The model takes care of the data and handles the basic behaviors of the application [4]. The view is a visual representation of the model, it handles the user interface and it does that by rendering the data from the model in a way that is convenient for the user interface. A view will have to know the semantics of the attributes of the model it represents. The controller represents the link between a user and the system, providing the user with input by arranging relevant views in appropriate places of the screen. It collects user input and handles calls to the model objects and view in order to execute fitting actions. [5]

It is mainly used when the connection between the view and the rest of the application is not always available. The idea behind it is to make a clear separation between the objects that model our vision of the real world (model), display objects which are the elements we see on the screen (view) and the model objects should work without any reference to the view objects, model objects being completely ignorant of the UI. [6]

A really good and simple example of the MVC pattern being used would be a simple web page which contains HTML and CSS. The model is the HTML which is basically text given as information to the reader. The CSS is the view because it adds style to the information; this style can be switched with any other style by modifying the CSS, without having to alter the information. The browser can be seen as a controller. It mixes and renders the HTML and CSS into the final output, the web page. It also collects the input from the user and sends it to the JavaScript code needed for the page to function correctly. [5]

# Web Applications using .NET

## 3.1 What is .NET

ASP.NET comes from “Active Server Pages .NET” and is a powerful framework that was developed by Microsoft and was first released in 2002. Since then, it has built up a lot of popularity among web developers and software development companies who choose this framework because of how powerful it is in creating rich web applications and web services. The framework uses the Common Language Runtime, or CLR, which gives its users the chance to write ASP.NET code in any language supported by .NET. [7]

## 3.2 Advantages of using .NET in a web application

ASP.NET can be used for creating a diversity of web applications such as: web based applications, social networking websites, business and corporate websites and not only. [7]

Here are a few of its advantages in web development:

1. The .NET framework is a very mature framework which is constantly maintained and upgraded by Microsoft. Since the latest version, .NET Core, it is also open source, which gives a lot of flexibility and balance. [8]
2. ASP.NET greatly reduces the amount of code needed for creating complex applications. This can improve the overall development speed and reduce its costs.[7]
3. The ASP.NET framework supports multiple languages [7], giving flexibility by allowing its users to choose a programming language that suits their interests best.
4. ASP.NET provides strong security and reliability because of its built-in Windows authentication system and allows custom configuration for every application. [7]
5. It drastically boosts overall performance by using native optimization, just-in-time compilation and smart caching technologies. [7]
6. It is constantly updated by Microsoft in order to keep up with the most recent technologies and requirements [7]
7. Visual Studio is a great bonus. It is considered by a large group of developers to be the best IDE in the world, but of course, this is subjective. Visual Studio has advanced code generation and completion capabilities [8]
8. Visual Studio’s code generation and completion features can be improved by using a tool called ReSharper which is available for some of its supported languages.
9. C# - a powerful programming language. Its syntax is very similar to Java but with some more functionality. Just like Java, C# is truly an object oriented language, the difference being that its target is the enterprise level development. [8]
10. Since .NET has been around for a lot of years, it has managed to build a generous user base all over the world. [8]

## 3.3 Most common .NET Frameworks for developing web applications

The options presented below provide the base of the application; these are back-end frameworks.

### 3.3.1 Web Forms

Web Forms populates the Web server with ASPX pages. The users have to request the pages and navigate through them. A Web Forms application can be turned into a hybrid, with focus on the SPA (or Single Page Application) model since ASPX provides support for Ajax features. Because direct HTML and JavaScript exposure was dangerous for both applications and developers back when Web Forms was introduced, Microsoft’s Web Forms kept HTML away while also proved to have great results. [9]

### 3.3.2 ASP.NET MVC

MVC populates the Web server with controller classes and (Razor) views. This framework provides access to the pages through more descriptive URLs that refer to actions specified in the controller instead of pages. Compared to other frameworks, the difference for the users of the application is small but there is quite a difference (both good and bad) for developers. MVC can be turned into a hybrid framework focused on the SPA model by adding Ajax features. [9]

### 3.3.3 Web API

Web-API is an Application Programming Interface (API) which receives and handles HTTP calls. It has architecture close to the ASP.NET MVC architecture since it also uses actions, model binding, controllers and routing. Web-API populates the Web server only with HTML and JavaScript files. Even if a SPA application will work significantly faster than a multiple page application, the SPA is more challenging to develop because it usually works best with more advanced front-end frameworks like AngularJS, AngularJS 2, React, EmberJS and so on. [9]

## 3.4 ASP.NET MVC versus Web-API

The major differences between the two are:

* A Web API layer is created from a number of controller classes but it is derived from a different base class ApiController. The HTTP GET methods represented on a controller can return raw data without having to explicitly pass the data from an ActionController container, as in ASP.NET MVC. [9]
* In the case of Web API, content negotiation is invisible to the developer most of the time. The data is returned as a JSON by default, however, the JSON formatter can be replaced with a general-purpose of type-specific custom XML formatter. The formatters are registered in the configuration of the application when it is created. [9]
* ASP.NET MVC has some of its aspects bound to Microsoft’s Internet Information Services (IIS) therefor the IIS is the best Web server choice for an MVC application. In the case of Web API, hosting can be done outside of IIS because it abstracts the hosting environment. [9]
* Web API aims at being main part of any Web application because of its capacity to provide a HTTP-reachable back-end, the tricky part being that it still needs an HTML-rendering engine. This engine can be server-side and be taken from ASP.NET MVC or even ASP.NET Web Forms but it can also be client-side and have the shape of a single page application. [9]

## 3.5 Conclusion

In conclusion, whichever ASP.NET technology or framework you choose, you can build a strong Web application with any of them. If you’re not so familiar with JavaScript technologies, Web Forms or MVC would be the best solution because of how easy it is to configure and use them out of the box. If you’re more familiar with JavaScript or you want to have a separate server-side or client-side HTML rendering engine while also using its ability to provide HTTP-reachable back-end, Web API would be the best solution.

# Application

## 4.1 Development Process

### 4.1.1 Software Requirements Specification

Specification of the software requirements is the first step when designing software applications. These specification requirements can be functionalities supported by the application, or constraints and particular cases that the application must offer a support for. There are only two types of requirements: functional, which define how the system cooperates with the environment, and nonfunctional, which define the usability and overall performance of the system. They may also include use cases which describe the software’s interactions with the user but the software requirements don’t offer any design suggestions or possible solutions to business or technology problems. [10]

The SRS is many times considered to be the parent document because every successive project management document like statements of work, software architecture specifications, design specifications, documentation plans and testing and validation plans are all related to the SRS. A well designed SRS should achieve four major goals: provide feedback to the customer, decompose the problem into component parts, serve as input to the design specifications and serve as the parent document for testing and validation strategies applied to the requirements for verification. [10]

The requirements of this application written in explicit language would look close to this:

“Create a web application which will allow a user (potential intern) to register and submit an application for different types of internships at a company. The application must support three different roles: General User (potential intern), Manager (the company’s Human Resources employees) and an Administrator (the general administrator of the application). The General User can submit an application to the available types of internships; the Manager can read, accept or reject applications submitted by the General User. The Manager should also be able to notify when the General User’s application has been accepted/rejected; the Administrator is the one that gives roles to other users (can assign a Manager role to a General user role for example) and he can also edit or delete users.

The application must run without lag or problems of any type. If something is loading in the background, the user must be notified through the UI of the application. The application must also be easily extendable.”

We can identify the functional and nonfunctional requirements from the above specification:

*Functional requirements:*

* Submit an application as a General User
* Accept/reject submitted applications as a Manager
* Change the role of a user (or any other data regarding the user) as an Administrator

##### *Nonfunctional requirements:*

* The application must be easily extendable

### 4.1.2 Use Cases and Scenarios

A scenario is a scene which portrays an interaction with a proposed system or user. It is also a tool used to describe a certain use of a proposed system by capturing the system viewed from the outside, e.g., a user. A use case is a task which needs to be performed by the actor with the help of the system, the actor is a user of a system belonging to a particular role and it can be a person or it can be an external system. It must be mentioned that the actor is a role not an individual and the actor must be beneficiary of the use case.

The next step is to identify the actors and use cases matching the functional requirements, by analyzing the above functional requirements. See illustration 4.1.

### 4.1.3 Analysis

More text here bla bla

Sequence diagram here

### 4.1.4 Design

Text here bla bla

Module diagrams here

## 4.2 The final result

Bla bla

Application screenshots here

# What next?

What can be improved and so on

# Conclusion

Problems and so on

# Bibliography

[1] Daniel Nations, October 17, 2016, “[Improve Your Understanding of Web Applications](https://www.lifewire.com/what-is-a-web-application-3486637)”

[2] Amos Ndegwa, May 31, 2016 “[What is a Web Application?](https://www.maxcdn.com/one/visual-glossary/web-application/)”

[4] Dave Jarvis, January 9, 2013, “[What is MVC?](https://softwareengineering.stackexchange.com/questions/127624/what-is-mvc-really)”

[5] Jeff Atwood, May 5, 2008, “[Understanding model-view-controller](https://blog.codinghorror.com/understanding-model-view-controller/)”

[6] Martin Fowler, July 18, 2006, “[GUI Architectures](https://martinfowler.com/eaaDev/uiArchs.html)”

[7] Administrator of QArea, August 8, 2015“[Advantages of using Asp.NET in Web Development](https://qarea.com/articles/advantages-using-aspnet-web-development)”

[8] Matt West, Quora, January 7, 2015, “[Advantages of using Asp.NET over other frameworks](https://www.quora.com/What-is-the-advantage-of-using-ASP-NET-over-other-web-application-frameworks)”

[9] Leonardo Esposito, Code Magazine, November/December 2013, “[Planning Web Solutions today](http://www.codemag.com/Article/1312051)”

[10] Donn Le Vie Jr., August 29, 2010, “[Writing Software Requirements Specifications (SRS)](https://techwhirl.com/writing-software-requirements-specifications/)”

[11] William Y. Arms, Cornell University, 2014, “[Scenarios and Use Cases](https://www.cs.cornell.edu/courses/cs5150/2014fa/slides/D2-use-cases.pdf)”