**BABEŞ-BOLYAI UNIVERSITY CLUJ-NAPOCA**

**FACULTY OF MATHEMATICS AND COMPUTER SCIENCE**

**SPECIALIZATION COMPUTER SCIENCE IN GERMAN**

**DIPLOMA THESIS**

**My vehichle management**

**Supervisor**

**Lecturer Diana Cristea, PhD**

**Author**

Cîrstea Ștefan-Daniel

**2022**

Contents

[Introduction 4](#_Toc95808194)

[Scope and motivation 5](#_Toc95808195)

[Technologies 6](#_Toc95808196)

[Server-side technologies 6](#_Toc95808197)

[Client-side technologies 7](#_Toc95808198)

[Application 8](#_Toc95808199)

[References 9](#_Toc95808200)

# Introduction

# Scope and motivation

Today, there are more applications that help people remember things, and many ways to store data about your daily activities. But some deadlines are easy to be forgotten especially when it takes more than one or two months to take place. Because of that, I had chosen to create an application which scope is to help people manage all their vehicle information.

Managing vehicles using mobile or web-based applications is not a new idea and it has been done before, an example of this kind is MOVCAR [1]. Because I know that time is very important in our days, anyone have more to do every day and because I am a driver, I like driving and I know I will always be a driver, my application is looking to help people manage their vehicles. Moreover, is looking to help them by extracting data from their documents.

There are more other applications that reminds people about their periodical mandatory activities or documents that need to be done in order to be safe as a driver. In Table 1 is shown a comparison between some of this apps and the app developed by me.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Application / Features | Deadline reminder | Fleet management | Fuel  Management | Data  extraction | Web  solution | Service  management | Cloud  solution |
| MvManagement | X | X | X | X | X | X | X |
| MOVCAR | X | X | X |  | X |  | X |
| MyCar | X |  |  |  |  |  | X |
| Car Alert | X |  |  |  |  |  |  |

Table 1.

# Technologies

In terms of technologies, it was a hard decision to be made. And for this application several options were available for the client-side and also for the server-side.

For choosing a backend technology more factors were taking into count and more frameworks. First option for this application as a server-side technology was for sure .NET because it is fast, can be deployed on any platform and I am most familiar with it, but has the disadvantage that you cannot create AI component that easy. Second option was Java with Spring framework, because it is open to OOP writeable code and is a fast-learning framework and is very used worldwide, but as well as .NET does not provide an easy AI solution. Another option was python because it is very easy to write, has support for AI, APIs are very easy to write, but as a weakness, it does not provide a strong typing convention.

When client-side solutions come in front the following options were considered: Angular because of it is a component-based solution, it has some support from ASP.NET Boilerplate and is most used web framework with .NET, but it will generate a lot of files in final, React.Js because of their growing over time and because it use components and has some awesome UI kits available, but the disadvantage is that development starts from scratch. And also, I considered using .NET Razor pages because it does not need two projects, can bind DTOs directly but it means that classic styling with CSS and JavaScript should be used.

After research, MvManagement (MvM) is using the latest version of technologies available for now. This application is designed on top of .Net Core 6 which was released on November 8, 2021, according to Microsoft official website [2] for the backend of the application. In terms of client-side technology MvM is using last version of the React.Js at the moment of writing, one of the most known and used java script framework taking into count the statistic done by Stack Overflow [3]. This statistic shows that React.Js is one of the most preferred and most used web frameworks and it will be clear why after next paragraphs.

## Server-side technologies

As mentioned, this project is built on one of the most used technologies all over the word. Why .Net? Because .Net is a widely used solution for developing applications, because it is fast, it is maintained by one of the word’s giant companies, Microsoft, allows asynchronous programming, is cross platform allows deploying on any platform not only on windows this concept is supported by Microsoft on their website [4] and last but not least .Net is open source that means that everyone can contribute add their knowledge in order tot get the best from this framework. In terms of programming languages .NET uses C# which is a solution that encourage all the OOP related principles as Encapsulation, Composition, Aggregation, Inheritance, Abstraction and so on.

The version of the .NET was an important decision to be done, and this MvManagement application is built on .NET 6 because it is an LTS (Long term support) release that means that it benefits from Microsoft support for more time. According to the book “C# 10 and .NET 6 – Modern Cross-Platform Development” by Mark J. Price [5] the LTS versions of .NET are supported since a new LTS version is released and Current versions are supported just for a period after a newer version come out.

MvManagement is built on top of ASP.NET Boilerplate framework [6] created by Volosoft and maintained by all community because it is an open-source framework, moreover, it also benefits by the support of the .NET Foundation. It is a Framework design on top of SOLID principles that is a very good solution for developing web application with actual practices and tools. What does it provide? ASP.NET Boilerplate (ABP) from my use and according to their documentation ABP offers a layered architecture based on DDD (Domain Driven Design), modularity, the possibility to build your own modules on top of their modules, multitenancy, a way of storing data about different entities on the same server named by Gartner Glossary [7] and also by the ABP official documentation. In additions, to this awesome programming features, ASP.NET Boilerplate provides a very useful documentation and a prompt, helpful GitHub community.

In terms of databased MvM remains stuck to Microsoft technologies and it uses Microsoft SQL DB. In order to get an efficient way of creating, mapping and using database models it is combined with Entity Framework 6 [8]. According to Microsoft Entity framework is an ORM (Object Resource Mapping) tool that allows CRUD operations without having to write SQL queries.

## Client-side technologies

For rendering the user interface MvManagement is using React.Js library. This library builds Single-Page application. According to the article “Single-page application vs. multiple-page application” written by Neoteric on medium.com [9] single-page application are web solutions that does not need re-rendering page in order to display new content. And the biggest advantage of using single-page applications is that all the resources (HTML, CSS, Script) are loading once in the app life. Also, this is a component-based JavaScript library, that means it builds component with their unique management context and this results in complex UI.

An advantage of this library is that it is open-source, and you can easily find support. Facebook from 2021 named Meta, decided to use React as an open to development project from 2013 since now, according to their GitHub release history [10]. And now is one of the most used and preferred by developers web framework taking into count the study done by Stack Overflow [3].

In addition to React.Js for this application is used an UI kit named Chakra UI [11]. This UI library was relatively recent released, in 2020, and is very popular among developers because is easy to use and is lined up with react actual standards. Chakra is offering a way of writing fewer and cleaner code, but also let developers extend component in order to get the best for them.

# Application

# References

|  |  |
| --- | --- |
| [1] | Movcar SRL, "MOVCAR Website," 2021. [Online]. Available: https://movcar.app/. |
| [2] | ".NET and .NET Core Support Policy," Microsoft, 2021. [Online]. Available: https://dotnet.microsoft.com/en-us/platform/support/policy/dotnet-core. |
| [3] | "Stack Overflow insights," Stack Overflow, 2021. [Online]. Available: https://insights.stackoverflow.com/survey/2021#most-popular-technologies-webframe. |
| [4] | "Microsoft - What is .NET?," Microsoft, [Online]. Available: https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet. |
| [5] | M. J. Price, "C# 10 and .NET 6 – Modern Cross-Platform Development," in *C# 10 and .NET 6 – Modern Cross-Platform Development*, 2021. |
| [6] | "ASP.NET Boilerplate," Volo Soft, [Online]. Available: https://aspnetboilerplate.com/. |
| [7] | "Gartner Glossary - Multitenancy," [Online]. Available: https://www.gartner.com/en/information-technology/glossary/multitenancy. |
| [8] | "Entity Framework 6," [Online]. Available: https://entityframework.net/. |
| [9] | Neoteric, "Mediu.com - Single-page application vs. multiple-page application," [Online]. Available: https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58. |
| [10] | facebook, "React repository," Meta, [Online]. Available: https://github.com/facebook/react/releases. |
| [11] | "Chakra UI," [Online]. Available: https://chakra-ui.com/. |