

ÖZYEĞİN UNİVERSİTY

FACULTY OF ENGINEERING

**CS400**

**SUMMER PRACTICE REPORT**

**Barış Karaer**

**S015497**

**INTERNSHIP COMPANY & DEPARTMENT:**

**Mirsis Information Technologies / Software Development**

**13.07.2020**

#### SUMMER PRACTICE REPORT

|  |  |
| --- | --- |
| **STUDENT** | |
| **Name** | **Barış Karaer** |
| **Internship Start Date** | **13.08.2020** |
| **Internship Completion Date** | **10.09.2020** |
| **Total Working Days** | **20** |
| **COMPANY** | |
| **Name** | **Mirsis IT** |
| **Department** | **Software Development** |
| **Address** |  |
| **SUPERVISOR** | |
| **Name** |  |
| **Title** |  |
| **Department** |  |
| **Phone** |  |
| **E-Mail** |  |
| **Signature** |  |

**DAILY WORK SUMMARY**

|  |  |  |
| --- | --- | --- |
| **DAY** | **DATE** | **WORK DESCRIPTION** |
| **1** | **13.08.2020** | Started the Service layer of the multi-layer project. Implemented the ProductService class, then the Service class. |
| **2** | **17.08.2020** | Finished the Service layer by implementing the Category Service class. By now I have finished all the layers for my project. I started the API section of this project. Updated the Startup file of the API and started implementing the CategoryController. |
| **3** | **18.08.2020** | Before fully implementing the controllers, I needed a DTO (Data Transfer Object) for every Entity in my project. I implemented the Category DTO for the controller. Added the AutoMapper library to map DTO objects to models or this case but in reverse. |
| **4** | **19.08.2020** | By finishing up the DTO file, I created a Mapper file which the project uses to change objects (DTO -> Model or Model -> DTO). After these implementations It was safe for me to start implementing the methods for the CategoryController. Finished the GetById and Save method. Then tested them by using Postman. |
| **5** | **20.08.2020** | Finished the Update and Delete methods for the CategoryController, then tested whether they worked or not using Postman. |
| **6** | **21.08.2020** | After finishing the GetWithProductsById, testing the method using Postman and handling some errors, I started implementing the ProductsController. Finished the GetById method for the products and tested it. |
| **7** | **24.08.2020** | Finished the Save/Update action methods for ProductsController and tested them with Postman. |
| **8** | **25.08.2020** | Finished the Delete/GetWithCategoryById action methods for ProductsController and tested them with Postman. |
| **9** | **26.08.2020** | Creating the Person Entity/Model and creating the migration to the database (MSSQL). PersonController and the action methods are implemented. |
| **10** | **27.08.2020** | Created the ValidationFilter and used this filter for the Category, Product controllers. This filter is used to handle Exceptions and create a certain JSON error. Tested this filter using Postman, created different exceptions for different attributes of these controllers. |

**Student’s Name: Barış Karaer Supervisor’s Name:**

**Student’s Signature: Supervisor’s Signature-Stamp:**

**DAILY WORK SUMMARY**

|  |  |  |
| --- | --- | --- |
| **DAY** | **DATE** | **WORK DESCRIPTION** |
| **11** | **28.08.2020** | Created the NotFoundFilter and used this filter for the Category, Product controllers. This filter is used to handle Exceptions (if a model is not found in the database) and create a certain JSON error. Tested this filter using Postman, created different exceptions for different attributes of these controllers. |
| **12** | **31.08.2020** | Handled errors globally. Created a custom ExceptionHandler, implemented it. Created a separate folder for this ExceptionHandler method to code using the best practices. |
| **13** | **01.09.2020** | Handled some global errors again and finished the API module of this project. Tested if all the methods are correctly running through Postman. Essentially, I finished the project, finished all the layers, the API and tested them. But I haven’t created an MVC application for this project. There are 2 ways of doing this. Using the layers, I already have or using the API. I will create the MVC project using the layers, I won’t have time for finishing MVC application using the API but will be finishing that after the internship is finished. |
| **14** | **02.09.2020** | Created the MVC project and a different branch for the GitHub repository for this project. Added the references from the layers I created (Core, Service, Data modules). |
| **15** | **03.09.2020** | Added the AutoMapper library for the MVC project. Created the DTO files for the project because we would be needing them for this module too. After adding the AutoMapper library implemented the Mapper folder for the conversion of these Models and DTO’s. |
| **16** | **04.09.2020** | Created the CategoriesController and referenced the Interfaces I will be working on in this controller. Implemented the Index method of the controller and Created View folder for the MVC project. |
| **17** | **07.09.2020** | Implemented the Index file for the Views/Categories folder. This file gets all the Category Data Transfer Objects and shows it in a web page. After this web page implementation, I created the buttons and routes for deleting, updating or creating a new category. |
| **18** | **08.09.2020** | Updated the CategoriesController for creating a category object. Then implemented the front end (View for this web page). Tested the implementation in MSSQL database and in the browser. |
| **19** | **09.09.2020** | Updated the CategoriesController for updating and deleting a category object. Then implemented the front end for updating the categories (View for this web page). Tested the implementation in MSSQL database and in the browser. |
| **20** | **10.09.2020** | Implemented the NotFoundFilter for the MVC .NET Core project. Finished the MVC project, tested it and finished my internship. All of the layers for the project, the API, and the MVC application is finished and uploaded in the GitHub version control system for further development. |

**Student’s Name: Supervisor’s Name:**

**Student’s Signature: Supervisor’s Signature:**

# Abstract

I have conducted my internship in Mirsis Information Technologies. I have earned experience in Full Stack technologies, used ReactJS, .NET Core and earned experience in multi-layer structure .NET Core projects. I have learned how to setup a web application using frameworks, building API’s, creating MVC web applications and creating databases using MSSQL.

# Introduction

The problem I worked on was a multi layered .NET Core project and an MVC application using these layers and the API. Then later on users would be able to create, delete or update these categories or products. Setting up a database, the coding environment for .NET and Postman were needed for solving the problem. I needed to learn how .NET Core projects were implemented and how to create a multi layered application using the best practices, coding in such a way that it is accepted worldwide. I also had to learn these best practices and apply them to this project. I had experience with C# coding, but I have never worked with these frameworks, worked with .NET Core or created an API of my own. I am already familiar with database concepts and relational databases so working with MSSQL was very easy and I quickly learned the workbench of MSSQL, I am also familiar with using Postman, but I was not very experienced. Furthermore, a new technique I learned was organizing my code in a multi structured layer so every module would do a specific task. I have learned how interfaces worked but I haven’t used them in a project before and I have learned about the MVC model that some applications used but I never got to experience creating an MVC application myself. In my .NET Core multi-layer project, I have used interfaces extensively and used it for a better and organized code.

# Company Description

Mirsis Information Technologies responds to the distinctive needs of different sectors and businesses in the field of information technologies. There are many research and development projects that Mirsis works on such as AI Chatbots, AI Medical Assistants, Smart HR Application and ATM Security. Also, Mirsis offers many services such as Turnkey projects, process consultancy and outsourcing. Mirsis has lots of references from variety of sectors like finance, insurance, telecom, IT and Health. For example, Acıbadem, IBM, Akbank, Yapı Kredi, Garanti BBVA are some of the references among many others which can be found on their website. The founders of Mirsis are Gül Düzgider (CEO) and Ece Kutlucan (Managing Partner). They created the company in 2008. Mirsis now has over 133+ software developers, 53+ analysis / test experts, 11+ project managers, 17+ database and system experts and 26+ technicians. I worked in the IT department where these projects are implemented and developed. My role was to learn different Front End, Back End web technologies and create a Web API, dashboard where users can store their personal information.

# .NET Core Multi Structured Web API

## Problem Statement

I have worked on a Web API which is built on the principals of a multi-layer structure. I have finished all the layers of the .NET Core project which consists of Core, Data and the Service layers. After implementing these layers, I have created an API that uses these layers. Used MSSQL as database and created a data table for this project. This project was based on the principals of a multi-layer structure and I created this project based on worldwide best principals that are used by many companies. After implementing the API, I have created an MVC application that uses the layers I have created and completed my internship.

The constraints for coding a multi-layer structure is creating different modules where these modules have separate tasks and linking them appropriately. This makes the problem much harder to solve because it takes time, but it is very organized and if an engineer codes their project based on these principals, other engineers will quickly understand and grasp what the code is about.

This is not a new problem for the company, my supervisor said this principal is widely used in other companies too where .NET Core is used.

Additionally, to CS300, I have finished the Web API and implemented an MVC application using the layer modules I have created during my internship.

Other national and international companies use this principal for creating their projects. Then they do unit tests for their projects and then deploy it.

## Tools and Techniques Used

I have used the C# language for using the .NET Core Framework. The purpose was to create a multi-layer structure project, an API and an MVC application. There are many other languages I could have done these principals with, but .NET Core is the main framework I should be using for creating API’s web applications because this framework is widely used around the globe. I have used the Auto Mapper library because I had to turn an object into a DTO (Data Transfer Object). Other than the Auto Mapper I have used the Entity Framework; this framework is being used because it organizes the models. In the MVC application I have used CSS, HTML for the View folder. I may have used other front-end options / programming languages / frameworks, but I wanted to do all this in .NET Core. I have already used React.JS framework for the CS300 timespan. For my purposes the front end was not a priority it just had to be functional enough for me to test the MVC that I created using .NET Core. This project does not use any hardware. This multi-layer structure technique is employed worldwide for better and cleaner code. The availability, usability and the readability of the code is far greater than other principals. Therefore, it is commonly used in other companies too.

## Detailed Explanation

First, I would like to talk about the API project. I began the project by updating the Startup file and creating controllers for my models. Then I found out that I needed to create DTO’s (Data Transfer Objects) for my project because it is not best practice to return models in functions. It is best practice to return DTO’s such that there wouldn’t be any data leak. All the required coding environments were ready for me to do this, I have already downloaded .NET Core requirements in the CS300 internship. I have implemented many methods for these controllers that would let users create / update / delete or get these models. After implementing these controllers and their methods within them for categories and products, I have implemented filters that would handle exceptions in the case of many error test cases. After these filters were implemented, I organized my code for applying to the rules of best principals and finished the API project. I have used Postman for testing the API. These API’s are normally tested with the application “Postman” because in development, developers need testing if their methods work while they don’t have an interface they can test on. After testing and confirming that all of the API’s methods work, I finish the API project.

Secondly, I have implemented an MVC application for the multi-layer structure project I worked on. I have not used the API for this project, which I will be working on soon, but I have used the Core, Data, Service layer codes that I have finished early on. I have created the Models and Mapping folders for this project. These folders are nearly the same as the API folders. Auto Mapper library was needed for this project too, so I downloaded it and created DTO classes for this project too. DTO classes were needed to get data from the database / creating / deleting / updating them. After creating the controllers, I have implemented the index page where that model can be seen in a table. These information’s are acquired through a GET method that gets the data from the database. Through this index page, this model can be created, updated or deleted through various buttons. Then I have created the methods and View pages for them. The only testing method was to run the application and check with the database. If the changes are made, then it would be successful. The MVC project was finished after testing all the methods and the last day I created a NotFoundFilter for the MVC application that checks if an ID is not found in the database during a query. Then the MVC application was finished.

## Results

The mixed ReactJS and .NET Core API project was successful. Since my solution is in my computer and can only be shown in my own computer, the code repository [1] can be found in the appendix.

The WebAPI project that is based on the multi-layer structure is finished, I also made an MVC application that approves the layers I created. I updated the repository [2] for the project and the layers, API, the MVC application can be found there.

# Conclusions

The variety of computer science courses I took from Ozyegin University helped a lot in this internship. CS 102 Object Oriented Programming course helped me a lot for the .NET Project I made in this internship. If it wasn’t for that course I wouldn’t have heard of interfaces and how a project would be created based on the Object-Oriented programming principals. Also, the CS 201 Database course I took really helped me learn relational databases. We learned My SQL in that course, but it is similar to MSSQL. That’s why I was familiar with the concepts. The in-class education assignments and the project assignment in my internship was very similar because I used the same principals, I learned in the courses I took. For example, I used Git for version control systems, and I tried to code very efficiently In a short timeframe. The CS 320 course project was very similar to this project. They were different in a way that there wasn’t any specific hardlines. As an example, in some course projects some of our professors provide a demo and want us to follow that demo to finish up a project, but in this internship I didn’t start from any demo and I planned my coding journey, then started programming. So far, I don’t think any information I learned can be used on any of my class works because I don’t think Ozyegin has any web development courses, but this internship helped me learn how companies start projects and how they develop their API’s. In the future, these skills might help me achieve the web development positions I would want to apply. This internship changed my career goals because in the past I was not aware of the API technologies but now I would like to be involved in researching and developing in this area of Computer Science. My internship was online due to Covid-19 and I couldn’t get to work in the offices of Mirsis but my supervisor helped me and guided me these past 20 days. It was fun working and learning new things. I would have wanted to work with other engineers and get feedback from them, but it wasn’t possible because of the pandemic.

# Appendix

Include relevant material such as catalogues, product specifications, papers

1. <https://github.com/bariskaraer/ReactJS-.NET-Project>
2. https://github.com/bariskaraer/UdemyNLayerProject.API

# References

Each information, figure, table, etc. that does not belong to you (has been found online, taken from some other document, etc.) **must** be referenced, or you risk being penalized due to plagiarism.