|  |
| --- |
| Kjell Coppin – Howest |

|  |
| --- |
| Plan of approach |

Topic:

Design Patterns in .NET

# Research Questions

What are the most important design patterns in programming, and how are they applied in C#?

Which use cases are these design patterns best used for in relation to scalability and performance?

# Origin

The idea originated from Sander (my internship mentor at Bestmix software), with a focus on researching design patterns within the BestMix software.

# Background Research

Explore basic patterns in programming.

Study creational patterns such as Singleton, Prototype, Builder, and Factory.

Investigate structural patterns like Façade and Proxy.

Analyze behavioral patterns including Iterator, Observer, Mediator, and State.

# Methodology

## Literature Review:

Read/watch “Design Patterns Elements of Reusable Object-Oriented Software” and other informative sources.

Look at criticisms and alternative viewpoints of this material.

Learn how all these patterns are implemented in c#. Go in depth to find best practices, benefits, common uses, … All to acquire the expertise to execute the analysis part.

## Analysis of BestMix Software:

If permitted, examine the BestMix software's codebase.

Find out the frameworks used and look into the design patterns that the framework implements by default. Then find out which design patterns have been used by the rest of the application.

Draw conclusions regarding the effectiveness and efficiency of the applied patterns.

Identification of Improvements:

Propose potential improvements in design patterns based on the analysis.

Consider how alternative patterns or adjustments could enhance performance or maintainability.

# Objectives

Provide insights into the efficiency of different design patterns in .NET applications.

Determine the suitability of various design patterns for different application scenarios.

Demonstrate the application of key design patterns in C# programming.

Offer recommendations for improving design patterns within the BestMix software.

# Timeline

## First Half of Internship:

Conduct extensive literature review.

Gain a solid understanding of basic and advanced design patterns.

## Second Half of Internship:

Analyze the BestMix software's codebase and identify utilized design patterns.

Evaluate the effectiveness and efficiency of the employed patterns.

Propose and discuss potential enhancements or alternative design patterns.

# Sources

## Already investigated:

Fireship - "10 Design Patterns Explained in 10 Minutes": [YouTube Link](https://www.youtube.com/watch?v=tv-_1er1mWI)

TechWebDots - "Design Patterns in C#": [YouTube link](https://www.youtube.com/watch?v=bzQeMqcgILY&list=PLBEm2Vv2nD-Ppk8U_LaR8wXl47kgCI8Dl)

## To Investigate Sources:

Design Patterns: Elements of Reusable Object-Oriented Software: [PDF Link](:%20https:/www.javier8a.com/itc/bd1/articulo.pdf)

Refactoring Guru: [Website Link](https://refactoring.guru/design-patterns)

TechWebDots - "Design Patterns in C#" Playlist: [YouTube link](https://www.youtube.com/playlist?list=PLBEm2Vv2nD-Ppk8U_LaR8wXl47kgCI8Dl)

Dofactory – “C# Design patterns”: [Website Link](https://www.dofactory.com/net/design-patterns)

# Conclusion

This plan of approach outlines the systematic process for investigating design patterns in .NET applications, with a specific focus on their efficiency, suitability, and application within the BestMix software. By following this plan, I aim to provide valuable insights and recommendations for optimizing software design through the effective utilization of design patterns.

*Side note: I mention the book “*Design Patterns Elements of Reusable Object-Oriented Software” *often in this document. During my initial research I noticed that most sources reference this book. I have read part of it and it seems like this book is by far the single most important source relating to design patterns in software.*