# Module evaluatie project

# Mijn Reizen

## Algemeenheden

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Naam** | Ayrton Polfliet | | | | | | | | |
| **Datum** | 15/06/2020 | | | | | | | | |
| **Tijdsduur** | De volledige module | | | | **Effectief tijdsduur** | |  | |  |
| **Oefening nummer** | B1-MO-04-01 | | | Naam | | | | **MijnReizen** | |
| **Repository naam** | | | | | | | | | |
| *B1-MO-03-MijnReizen* | | | | | | | | | |
| **Projecteigenschappen** | | | | | | | | | |
| **Project/Solutionnaam** | | B1-MO-03-ReisApplicatie | | | | | | | |
| **Type** | | WPF | Console | | | Mobile | Webapp | | |
| **Indienen** | | Toledo | Git Hub | | | **Software pattern** | MVVM | | |

Lees aandachtig de opdracht onderaan de opgave

## Situatieschets

Je gaat veelvuldig op city trip en bezoekt daar de bezienswaardigheden. Het zou handig zijn wanneer er op een heel eenvoudige manier alle info van deze city trips en bezienswaardigheden op een heel eenvoudige manier kan beheerd en geraadpleegd worden.

Daarom is er besloten op een applicatie te ontwerpen welke het mogelijk gaat maken al deze informatie te beheren een te raadplegen. In eerste instantie als een desktopapplicatie. Nadien volgen een mobile en een webapplicatie.

De applicatie wordt ontworpen volgens een drie lagen architectuur. De desktop en de mobile versie van de app zal eveneens het MVVM architectural pattern implementeren.

De desktopapplicatie zal de nodige gegevens beheren via bestanden in een csv en of json formaat. De Webapplicatie zal de gegevens beheren in een database en beschikbaar stellen via REST-api.

## A screenshot of a cell phone Description automatically generatedUML-Classdiagram

Hieronder is het Classdiagram te vinden met de nodige klassen

## Indienen

1. Elke zondagavond moet het git repository gesynchroniseerd op github, zodat ik de voortgang continue kan volgen.
   1. Elke nieuwe grote wijziging aan de applicatie wordt in een afzonderlijke branch geïmplementeerd.
   2. De commentaar van de commit moet heel duidelijk en kort beschrijven wat er is gewijzigd.
   3. Via de Wiki pagina van de GitHub repository bespreek je welke branch en hou je per dag bij wat je hebt gewijzigd.
   4. Het Word-document met de opgave, moet aanwezig zijn in de master branch van je repository en correct ingevuld en dezelfde naam krijgen als je Visual Studio solution.
2. Er zullen 2 upload momenten zijn op Toledo, waar je de solution als een zip-bestand upload. Waarvan de tweede upload moment op Toledo de afgewerkte versie zal bevatten.

## Opdracht

1. Creëer een correct benoemde Visual Studio solution welke de Presentation en Business Logic Layer bevat.
2. De applicatie moet ontwikkeld worden volgens MVVM architecturaal patroon.
3. Ontwerp de nodige businessmodellen gebaseerd op het hierboven vermelde UML-classdiagram. Voor het implementeren van de verschillende businessmodellen worden de .NET properties gebruikt en krijgen het meest optimale datatype.
4. Van de methoden Save en Search moet enkel de structuur worden geïmplanteerd en deze moeten de **Not Implmented Exception** opwerpen met een correct bijhorende foutmelding.
5. De businessmodellen moet voorzien worden van de correct validatie waar nodig.
6. Voor het ontwerpen van de nodige schermen is er volledige vrijheid, alleen moeten de volgende zaken gerespecteerd worden:
   1. de gebruiksvriendelijkheid primeert
   2. voor elk veld het meest geschikte control kiezen

# Evaluatie criteria

|  |  |
| --- | --- |
| Github Repository & Wiki |  |
| Correct Visual Studio Solution |  |
| MVVM correct geïmplementeerd |  |
| Correcte validatie |  |
| Userinterface |  |
| Beheer van de data |  |
| Werking |  |
| Eigen inbreng |  |

# Feedback