

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA



Client analysis

Jim Marques Duarte

Mestrado em Informática

Trabalho de Projeto orientado por:
Prof. Dr. Luis Manuel Moniz

2021

Acknowledgments

I would like to thank all the people who contributed to the success of my project and who helped me in writing this thesis, without whom I would not have made it through this project.

Many thanks to Professor Luis Manuel Moniz, my thesis supervisor, for his support, invaluable advice, encouragement, patience, and availability. He was able to guide me in the construction and writing of my academic paper through judicious advice, which contributed to my reflection.

I would like to express my gratitude to the entire CIU department for their kindness, professionalism, availability, and respect during the course of the project. A special thanks goes to Mike Kieffer, Giuseppe Cardinale, Zaffaroni Gaia, and Flavius Paler for their support, patience, availability, for sharing their knowledge with me, and for never leaving me hanging whenever advice was needed.

I would like to express my gratitude to my family and friends who have provided me with moral and intellectual support throughout my journey.

Finally, I would like to give a massive “thank you” to my company supervisor, Santiago Rivas for all the patience, availability, support, and guidance in the writing of my thesis and throughout the whole development of the project. Thank you for sharing your knowledge and expertise with me and for having been the pillar I could rely on during the whole project.

Glossary

CIU	Client Intelligence Unit
B.I.R.T	Business Intelligence Reporting Tool
PSA	Portfolio Service Annuities
PSR	Portfolio Service Renewals
IP	Intellectual Property
TM	Trademark
SQL	Structured Query Language
SSMS	SQL Server Management Studio
CI/CD	Continuous Integration/Continuous Delivery or Deployment
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
AD	Active Directory
RDBMS	Relational Database Management System
DBA	Database Administrator

Abstract

Data visualization existed long before all modern data visualization tools were invented, initially used on maps that feature markers of land, cities, roads, and features. As time went on, the need for more accurate physical mappings and measurements grew, data visualization also had to grow accordingly, continuing to enhance the latest existing data visualization solution.

However, creating a visualization is not as simple as throwing the "information" element of an infographic on top of a graphic to make it look nicer. A careful balancing act between form and function is necessary for effective data display. The most basic graphic may be too uninteresting to look at or may send a strong message; the most impressive representation may completely fail to convey the proper idea, or it may say too much. Information and images must complement each other, and combining strong analysis with good storytelling is an art.

To summarize, the purpose of data visualization tools is to make sense of data, and how well that data is integrated into the visuals depends drastically on the tool used and the person using the tool. Every company must incorporate at least one data visualization tool to make significant gains in its crucial areas of operations.

Client analysis in correlation with behavioral analysis is one of the key aspects in companies. Analyzing clients may be one of the most important business topics inside every company due to this being a critical part of their business model and their marketing plan. Giving a deeper insight into the trends and patterns that occur in one of their key products, helps the company adapt and take into consideration their clients' needs in more detail and further improve their IP services. In particular, the company, on which this work is based, considers their clients also as partners, they need to be able to have some access to a huge diversity of data regarding the trademarks that they have and are being taken care of.

Examples would be: knowing the applied fees, the payment and instruction dates, the best country law modeling available in the market, trends, etc.

This thesis provides insight on how the work carried out during the project brings updates to the department, improving existing tools such as the Pentaho portal and the Web2Py portal. In addition, a merged implementation of the portals facilitates the maintenance of users of both portals, facilitating the life of the department through the automation of some procedures. Finally, the study of data visualization tools was done so that some of

the main implementations were presented and discussed among the relevant people in the department. This thesis focuses on the trademark department and allowed me to acquire a lot of knowledge about how to work as a team, individually, manage my time, set priorities, direct relevant people and be critical in matters related to the project.

Keywords: Web2Py portal, Pentaho portal, visualization tools, analytical modules, Portfolio Services Renewals

Resumo

A visualização de dados já existia muito antes de todas as ferramentas modernas de visualização de dados serem inventadas, inicialmente usadas em mapas que apresentam marcadores de terras, cidades, estradas e recursos. Com o passar do tempo, a necessidade de mapeamentos e medições físicas mais precisas cresceu, a visualização de dados também teve que crescer de acordo, continuando a aprimorar a solução de visualização de dados mais recente existente.

Agora no mercado, há uma variedade de ferramentas de visualização de dados de código aberto, parcialmente aberto e pago com uma variedade de recursos disponíveis para extrair e exibir todos os tipos de dados fechado.

Além disso, todas as recentes e poderosas ferramentas de visualização de dados disponíveis, permitem a ligação a diferentes fontes de dados, como bancos de dados SQL, Excel, Json, CSV, entre muitos outros. As suas visualizações poderosas e reativas são uma grande atualização em termos de análise em muitas grandes empresas.

Aliás, o uso de ferramentas de visualização de dados mais poderosas e recentes beneficia os analistas para identificar padrões, representando dados complicados de uma forma muito legível, e torna os usuários mais cientes em certas partes dos dados que requerem atenção para fazer seus negócios avançarem tomando melhores decisões de negócios. Encontrar erros também é uma vantagem, pois permite que os usuários encontrem visualmente os erros de dados, identificando as áreas que indicam os sinais de alerta. A tomada de decisão por meio da análise de visualizações é melhor, pois humanos são capazes de processar visuais muito melhor do que formulários tabulares de relatórios de dados, o que também permite o crescimento da empresa.

À medida que a "era do Big Data" se acelera, a visualização está se tornando uma ferramenta cada vez mais importante para compreender os bilhões de linhas de dados criados no dia à dia. Como humanos estamos programados para nos envolvermos com histórias. Na forma mais básica, os dados são tudo, menos uma história - na verdade, dados na sua forma mais básica são exatamente o oposto.

A visualização de dados ajuda a contar histórias, transformando os dados em um formato mais compreensível mostrando padrões. Uma boa visualização conta uma narrativa reduzindo o ruído dos dados e enfatizando os fatos mais importantes.

A criação de visuais é fácil e muito interativa na maioria dos casos, não necessitando de nenhuma ou quase nenhuma experiência de programação. A visualização de dados

amplifica o efeito de suas mensagens para seu público-alvo e oferece resultados de análise de dados da maneira mais atraente possível.

No entanto, não é tão simples quanto jogar o elemento "informação" de um infográfico no topo de um gráfico para torná-lo mais agradável. Um ato de equilíbrio cuidadoso entre a forma e a função é necessário para a exibição de dados eficaz. O gráfico mais básico pode ser muito desinteressante para ser visto ou pode enviar uma mensagem forte; a representação mais impressionante pode falhar completamente em transmitir a ideia apropriada, ou pode dizer muito. As informações e as imagens devem se complementar, e combinar análise marcante com boa narrativa é uma arte.

Para resumir, o objetivo das ferramentas de visualização de dados é dar sentido aos dados, e quão bem esses dados são integrados aos visuais depende drasticamente da ferramenta usada e, da pessoa, que usa a ferramenta. Cada empresa deve incorporar pelo menos uma ferramenta de visualização de dados para obter ganhos significativos em suas áreas cruciais de operações.

A análise do cliente em correlação com a análise comportamental é um dos aspectos-chave nas empresas. Analisar clientes é um dos tópicos de negócios mais importantes dentro de qualquer empresa, por ser uma parte crítica de seu modelo de negócios e plano de marketing. Dar uma visão mais profunda das tendências e padrões que ocorrem em um de seus principais produtos ajuda a empresa a se adaptar e levar em consideração as necessidades de seus clientes com mais detalhes e melhorar ainda mais seus serviços de IP. Em particular, a empresa na qual se baseia este trabalho considera os seus clientes também como parceiros, estes precisam de poder ter algum acesso a uma grande diversidade de dados sobre as marcas que possuem e estão a ser geridas.

Os exemplos seriam; conhecer as taxas aplicadas, estudar instruções tardias, as datas de pagamento e instrução, a melhor modelagem de legislação do país disponível no mercado, tendências, etc.

A unidade de inteligência do cliente (CIU) começou recentemente a analisar extensivamente os dados disponíveis para oferecer melhores produtos aos seus clientes. No entanto, embora tenham realizado um trabalho extensivo em algumas áreas (especialmente nos seus principais produtos mais vendidos), não podem negligenciar os seus outros serviços.

Em particular, depois de examinar que áreas poderiam ser melhoradas, chegámos à conclusão de nos concentrarmos no departamento de Renovação de Marcas produzindo meios analíticos para os utilizadores a fim de me familiarizar com as tecnologias utilizadas. Conhecer o comportamento e os desejos dos clientes é crucial para a qualidade do negócio, não só para lhes prestar um melhor serviço, mas também para poder adaptar-

se a um cenário global em constante mudança que se tornou ainda mais desafiante com a pandemia de 2020.

Outra questão importante que foi detetada foi que, devido à complexidade e flexibilidade da unidade, o departamento estava a apoiar dois portais para fins analíticos. Portanto, os utilizadores deviam ligar-se a dois portais a fim de aceder a todas as funcionalidades, em vez de terem todas as funcionalidades/módulos analíticos num único portal. Como seria de esperar, isto vem com a questão comum de ter uma forma não harmonizada no tratamento da segurança e na criação de relatórios.

Além disso, os desenhos, estrutura de codificação, módulos e controlo de versões diferem entre os dois portais, não há consistência no desenho e em como utilizar os módulos para a parte analítica. Ambos os portais, usando tecnologias diferentes, não tornam este problema mais fácil de resolver. Encontrar uma maneira de fusionar os dois portais e facilitar a criação de funções, criação de contas e ter uma interface onde só é necessário de fazer o login uma vez para poder explorar todas as ferramentas analíticas, é o objetivo da terceira parte do projeto.

Como ponto final, no que se refere à visualização analítica, a unidade está na realidade a codificá-la usando bibliotecas Python ou utilizando o leque limitado de possibilidades, que o Pentaho oferece. Fornecer aos utilizadores uma análise sobre os benefícios e como integrar um software de visualização de dados mais complexo e sofisticado ajudá-los-ia sem dúvida a planear um futuro com uma gama mais ampla de ferramentas para melhorar a eficiência dos relatórios e das visualizações. Consequentemente, três ferramentas de visualização de dados poderosas e muito usadas foram testadas e comparadas, Power BI, Alteryx e Qlik. Essas ferramentas foram avaliadas definindo primeiro uma lista de funcionalidades essenciais e, em seguida, pesquisando e testando parcialmente se as ferramentas oferecem suporte a essas funcionalidades. Além disso, foram feitas várias demonstrações das aplicabilidades requeridas das ferramentas à equipa da CIU, apoiando-se mais no Power BI, sendo essa a ferramenta para a qual já está a ser paga uma licença PRO para cada utilizador.

Resumindo, esta tese fornece uma visão sobre como criar módulos analíticos em Web2Py e Pentaho para analisar clientes e o seus comportamentos, sendo elas duas tecnologias open source. A parte analítica da tese gira em torno dos clientes de marcas, uma vez que é a parte menos desenvolvida analiticamente da empresa. Além de uma descrição clara dos módulos analíticos criados, ela também contém uma visão clara sobre como 2 portais existentes usando tecnologias diferentes são unidos em um, unindo as suas ‘interfaces’ para fazer parecer ao utilizador que está só a navegar num único portal, na pesquisa dos seus módulos analíticos. Um usa a tecnologia Web2Py programado em Python e baseado no MVC e o outro sendo o Pentaho programado em Java, que usa JS para os seus módulos

analíticos e está ligado à ‘software’ de relatórios B.I.R.T. Por fim, a tese termina com um estudo sobre diferentes ferramentas de visualização poderosas para substituir módulos e gráficos existentes e sobre como eles poderão ser integrados na interface fundida.

O trabalho realizado durante o projeto traz atualizações para o departamento, aprimorando as ferramentas existentes como o portal Pentaho e o portal Web2Py. Além disso, uma implementação fundida dos portais facilita a manutenção dos usuários de ambos os portais, facilitando a vida do departamento por meio da automação de alguns procedimentos. Por fim, o estudo sobre ferramentas de visualização de dados foi feito de forma que algumas das principais implementações foram apresentadas e discutidas entre as pessoas relevantes do departamento. Esta tese gira em torno do departamento de marcas e permitiu-me adquirir muitos conhecimentos sobre como trabalhar em equipa, individualmente, administrar meu tempo, definir prioridades, direcionar pessoas relevantes e ser franco nos assuntos relacionados ao projeto.

Palavras-chave: Portal Web2Py, Portal Pentaho, Ferramentas analíticas, Visualização analítica

Content

Chapter 1: Introduction	1
1.1 Motivation	1
1.2 Objectives	2
1.3 Host institution	3
1.4 Context of work	3
1.5 Contribution.....	4
1.6 Structure of the document.....	6
Chapter 2: Concepts, definitions, and current structure.....	9
2.1 Existing work/Related tools.....	9
2.1.1 Web2py portal.....	9
2.1.2 Pentaho portal.....	10
2.1.3 B.I.R.T. tool.....	11
2.2 Other tools used.....	13
2.2.1 Microsoft SQL server.....	13
2.2.2 SSMS.....	13
2.2.3 Jupyter Notebook	14
2.3 Pipeline and deployment management	15
2.3.1 Git.....	15
2.3.2 Azure board and project management.....	15
2.4 Conclusion.....	17
Chapter 3: Analysis and planning	19
3.1 Deliverable 1 – Late instruction counter	19
3.2 Deliverable 2 – TM Client report	20
3.3 Analysis of the problem (Merging of portals).....	20
3.3.1 Fishbone diagram – Problem.....	20
3.3.2 Workflow of portal development – Login creation.....	21
3.3.3 Workflow of portal development – Creation of a report.....	22

3.3.4	Workflow of portal usage – Generation of a report	24
3.3.5	User role selection.....	25
3.4	Proposing a new solution.....	26
3.4.1	Deliverable 1 – Late instruction chart	26
3.4.2	Deliverable 2 – TM Client report.....	27
3.4.3	Merging of portals.....	27
3.4.4	Workflow of portal development – Login creation(new)	28
3.4.5	Workflow of portal usage – Generation of a report(new).....	29
3.4.6	User role architecture	29
3.5	Study on data visualization tools	30
3.5.1	Power BI.....	30
3.5.2	Alteryx.....	30
3.5.3	Qlik.....	31
3.5.4	Analysis of the tools.....	32
3.6	Summary and conclusion.....	33
Chapter 4:	Implementation.....	37
4.1	Development process.....	37
4.1.1	Deliverable 1 – Late instruction chart	37
4.1.2	Deliverable 2 – TM client report.....	41
4.1.3	Deliverable 3 – Merging of portals	47
4.1.4	Deliverable 4 – Study on visualization tools.....	49
4.2	Testing process	49
4.2.1	Deliverable 1 – Late instructions chart	49
4.2.2	Deliverable 2 – TM Client report.....	50
4.2.3	Deliverable 3 – Merging of portals	50
Chapter 5:	Results	53
5.1	Validation process	53
5.1.1	Deliverable 1 – Late instruction chart	53
5.1.2	Deliverable 2 – TM Client report.....	55

5.1.3	Deliverable 3 – Merging of portals	57
Chapter 6:	Conclusion.....	61
6.1	Difficulties encountered	61
6.2	Future work.....	62

List of figures

Figure 2.1 – Web2Py portal	10
Figure 2.2 – Pentaho portal	11
Figure 2.3 – B.I.R.T. tool.....	12
Figure 2.4 – SSMS example	14
Figure 2.5 – Jupyter Notebook example	15
Figure 2.6 – Azure board example.....	16
Figure 2.7 – Agile methodology sprints.....	16
Figure 2.8 – CI/CD.....	17
Figure 3.1 – Fishbone diagram: causes & consequences.....	21
Figure 3.2 – Login creation (old - Swimlane diagram).....	22
Figure 3.3 – Report creation (Swimlane diagram).....	23
Figure 3.4 – Report generation (old - Swimlane diagram)	24
Figure 3.5 – User role query Web2Py (old).....	25
Figure 3.6 – User role query Pentaho (old & new).....	25
Figure 3.7 – Late instr. chart draft	26
Figure 3.8 – TM client report draft	27
Figure 3.9 – Merged portals draft	28
Figure 3.10 – Login creation (new - Swimlane diagram).....	28
Figure 3.11 – Login creation (new - Swimlane diagram).....	29
Figure 3.12 – User role query (new).....	29
Figure 3.13 – Power BI example	30
Figure 3.14 – Alteryx example	31
Figure 3.15 – Qlik example.....	31
Figure 4.1 – Connection class usage.....	38
Figure 4.2 – Jupyter Notebook live Code - Late instr. chart.....	38
Figure 4.3 – HTML code for new Card	39
Figure 4.4 – Web2Py portal - Late instr. Counter card.....	40

Figure 4.5 – late_instructions_counter.py - Web2Py form.....	40
Figure 4.6 – late_instructions_counter.py - Return for view	40
Figure 4.7 – LateInstructionCounter.html - Passed parameters,controller to view	41
Figure 4.8 – B.I.R.T. tool - Report creation objects	42
Figure 4.9 – B.I.R.T. tool - Client list report layout	42
Figure 4.10 – B.I.R.T. tool - Enter client list report parameter.....	43
Figure 4.11 – B.I.R.T. tool - Client list report data set creation	43
Figure 4.12 – B.I.R.T. tool - Data transformation JavaScript.....	43
Figure 4.13 – B.I.R.T. tool - Client list report hyperlink options	44
Figure 4.14 – B.I.R.T. tool - Client report layout	45
Figure 4.15 – B.I.R.T. tool - Data Sets	45
Figure 4.16 – B.I.R.T. tool - Invoice printed data set	46
Figure 4.17 – B.I.R.T. tool - Data Set and value selection for field	46
Figure 4.18 – B.I.R.T. tool - Expression Builder.....	47
Figure 4.19 – Pentaho portal - CIU tab.....	47
Figure 4.20 – Web2Py portal - Bypass login script	48
Figure 4.21 – Merged portals - Block Web2Py access from outside Pentaho.....	48
Figure 5.1 – Live version Late Instr. Counter (web2py portal)	54
Figure 5.2 – Live version TM Client report - Client selection (Pentaho portal).....	56
Figure 5.3 – Live version TM Client report - Report (Pentaho portal).....	57
Figure 5.4 – Merged portals.....	58

List of tables

Table 3.1 – Defined features required for the data visualization tools	32
--	----

Chapter 1:

Introduction

This introductory chapter presents the motivations behind this study, the objectives to be met with this project, and introduces the company that proposed this project and hosted me. Finally, this chapter finishes with the context of work, the contributions brought by the finalized work, and the description of the report in the distinct chapters.

1.1 Motivation

Defining and analyzing the clients are a must in big international company working with clients from all over the world. In order to be able to adapt services, the stored client data needs to be analyzed and taken into account when treating clients in order to provide the best service possible.

The Client Intelligence Unit which is the department that currently handles most of the internally stored data and provides internal services throughout the departments, has recently started extensively analyzing the available data to offer their clients better products. However, even though they have done extensive work in some areas (especially in their top-grossing products), they cannot neglect their other services.

In particular, after examining which areas could be improved, it was concluded that the focus of this project should be on the trademarks. One of the tasks of the trademark department is to manage their clients' trademarks, provide portfolio lists and also renew trademarks in due time freeing their clients from administrative workload, among many other services related to trademarks.

Knowing the behavior and desires of the client is a crucial aspect of the business quality, not only to provide them a better service but also to be able to adapt to an ever-changing global scenario that has become even more challenging with the 2020 pandemic.

Additionally, another major issue that was detected was that due to the complexity and flexibility of the unit, the department was supporting two portals for analytical purposes. However, both portals are unique in their architecture and are using different technologies to run.

Therefore, the users must connect to two portals in order to access all the available features instead of having all the analytical features/modules in one portal. As one might expect this comes with the common issue of having an unharmonized way of handling security and creating reports. Furthermore, the designs, coding structure, modules, and version controlling differ between the two portals. There is no consistency on the design and on how to use the modules for the analytical part.

As a final point, in what refers to the analytical visualization [10], the unit is hardcoding it using python libraries or using the limited range of possibilities that Pentaho offers. Providing the stakeholders a deep analysis on the benefits and how to integrate a more complex and sophisticated data visualization [40] software would undoubtedly help them plan towards a future with a wider range of tools to improve the efficiency of the reports and visualizations.

1.2 Objectives

The objective of this project is not only to upgrade the existing data analytical features and tools by creating and adding new analytical features to the existing portals, but to additionally simplify the access to the reports and dashboards by merging both portals in one. Moreover, the intention as well is to explore possible enhancements to create a state of the art reporting architecture.

Having two portals for analytical purposes for the same department requires a few deliverables for a fluent unfolding of the project. After revising the current structure, four main points were identified, that needed to be improved and analyzed in order to create a better user experience and prepare the department for a new state of the art data and analysis architecture.

The following objectives were defined:

- The creation of a data visualization module for the Web2Py portal
- The production of a data visualization module for the Pentaho portal
- The implementation of a merging of both portals (Web2Py & Pentaho)
- A study of different powerful data visualization tools

The first and second deliverables were not only to get a way of getting acquainted with the technologies used in the department but also to provide reporting tools that would offer deeper insights into the business.

For this purpose, deliverables one and two consist in applying existing analytical methodologies already used in the analytical part of existing products and new methodologies into another of their key products, which is the trademark renewals department. Therefore, enhancing the existing Web2Py and Pentaho portals by creating

two new analytical features, one for each portal, were defined as deliverable one and two. These first two deliverables will bring knowledge on how both portals are working, the methodologies used, and how their respective security is handled. Whereas the first analytical module was the creation of a module allowing users to check the late instructions received for specific clients and additionally check the amount the firm gets from these late instructions and the second one was the creation of a client report, with relevant client information that will be available for the user upon searching for a specific client. The relevant information will be displayed as a portrait for the user, describing the user in that specific report, containing information like “Active client?”, “Nr. Of Active cases”, “Payment behavior”, among other information.

The third deliverable, which is the most crucial part of the project, is merging both portals into one, keeping in mind that two completely different portals should be merged.

Finally, also contributing to the main objectives of the project, deliverable four is the study of three data visualization tools, that could further enhance or possibly replace both portals.

1.3 Host institution

Dennemeyer is an intellectual property consulting company offering trademark renewal services in addition to patent annuity services among many other services related to patents and trademarks. Being a “full-service provider” for global IP management since 1962, the firm has gotten a reputation for raising the bar for service in the IP business, and is engaged in finding new and better ways to secure and protect intellectual property rights throughout the world.

Dennemeyer began as a patent law firm but quickly expanded into a comprehensive full service provider, handling the whole IP management chain. The Luxembourg-based firm has grown into one of the world's top IP service providers thanks to a series of innovations. Currently, this international firm has over 20 offices around the world and offers more than 35 products in order to ensure that clients get all the possible solutions for their intellectual property rights.

1.4 Context of work

The project is a development, database, and research-related project. Needing the development of Python code for the creation of the chart for the web2py portal, JavaScript [14] for merging of the portals part of the project, and SQL for the ETL of data.

In terms of database, we hosted the information in a Microsoft SQL server, that had role-level security and ensured that we could query it through python and the Business Intelligence reporting tool (BIRT).

Lastly, a study in terms of studying new solutions for portal alternatives which support existing portal features used in the existing portals.

1.5 Contribution

Since the general objective of the project was to enhance and optimize existing solutions and study new solutions, two new modules were developed, one for each studied portal, the merging of both portals was implemented and a study on new data visualization solutions was done.

- ***Web2Py - Late instruction counter:*** Since the first objective was the enhancement of the web2py portal to get familiar with it, a data analytical module was created inside the CIU portal that allows the user to analyze the behavior of trademark clients according to late instructions in a specific time range. In that specific module, the user can search for a trademark client and select the time frame according to the range he wants to analyze the number of late instructions from. In addition to this, underneath the chart, the module also generates a table with the chosen range for a specific client with the according number of late instructions in that range and the amount of late instructions fees generated in that range.
- ***Pentaho – PSR BD INFO MAIL:*** For the second deliverable, the same principle as in the first deliverable was used. To get familiar with the Pentaho portal, the creation of a new data analytical module using the Business Intelligence reporting tool (BIRT) in eclipse, in this case, a report creation for Pentaho portal was the objective. However, this second module is a report containing the most important information about a client. This TM client report is generated after running the module and selecting the specific client we want information from and the date range for the client-specific information. In addition, the report is in an easily readable and printable A4 format. Just like the module in the first deliverable, this second deliverable extends the possibility to analyze trademark clients in more detail, since previously no similar report for trademark clients existed.
- ***Merging of portals:*** Due to the fact that there are two portals, a third objective was to find a solution to combine the two portals, whereas both solutions are open-source software; Web2py being a free web framework programmable in Python and Pentaho being a Business Intelligence software which allows data preparation, access, and

analytics. By concluding the two previous modules, enough knowledge has been acquired to study a way to merge both interfaces.

The merged implementation of the portals now allows the user to quickly navigate between portals, giving the sensation that he is navigating only through one portal, since the user only needs to log in once, to be able to profit from all the analytical tools available. Consequently, this provides the user with a more fluent experience having all the analytical tools provided by the department under one interface.

- ***Data visualization tools study:*** In order to help in a future improvement in the efficiency of the reports and visualizations for the data analytics in the department, a fourth deliverable came with the analysis of a more robust visualization tool and their integration into the existing portals or the replacement of part of the portals with a new tool.

The data visualization tools study report stays in the same area as the previous deliverables, taking into consideration everything that has been learned from the previous deliverables and bringing insight into three selected powerful data visualization tools that may be able to upgrade the portals even further or even replace them. The tools studied were Power BI, Alteryx, and Qlik. A later implementation of such a powerful tool may allow the department to take on heavier and more complex requests with more ease

This project brought a number of upgrades to the existing solutions, as well as a study on a potential replacement of the current solutions. Due to the fact that Dennemeyer cannot neglect analytics related to the trademark department, the new analytical web2py module and the new Pentaho report are a big upgrade. Data analysts will be able to quickly see the income for late instructions for specific clients and a new report for a quick insight on a specific client, basically showing a summary of the client with their most relevant information.

The portal merging also benefits the CIU department in terms of security handling and user creation. The security and user creation are not handled twice anymore through Web2Py and Pentaho but only once through newly defined role mapping from Web2Py and Pentaho roles. New users available in the active directory (AD) are now queried every Monday and automatically created in the respective external Pentaho and Web2Py databases with the roles also queried from the AD. The user now only needs to log in once into the Pentaho portal and has access to all the analytical tools available from both

portals, additionally lessening the burden of switching between distinct portals and having to log in twice.

And finally, the last deliverable, which was the study of visualization tools, benefits the company by giving them a detailed study on how powerful different visualization tools are, and based on a list of functionalities, if they support the desired functionality or not.

1.6 Structure of the document

The remaining chapters of this document are structured in the following form:

- Chapter 2 – Focuses on related work and the state of the art, comparing the various existing technologies and approaches in addition to introducing and clarifying some relevant concepts.
- Chapter 3 – Refers to the planning of the project, describes methodologies, meetings, resources, deadlines, and risks.
- Chapter 4 – Describes aspects related to project implementation such as code organization, developed features, and test execution.
- Chapter 5 – Focuses on the obtained results for the different deliverables, their importance on the project, and their validation.
- Chapter 6 – Concludes the document, verifying whether the outlined objectives have been achieved, identifying the main obstacles and the future perspective.

Chapter 2:

Concepts, definitions, and current structure

Data visualization already existed way before all the modern data visualization tools were invented, initially used in maps presenting land markers, cities, roads, and resources. As time passed, the necessity for more accurate mappings and physical measurement grew, data visualization also had to grow accordingly by continuing to enhance the existing latest data visualization solution.

Now on the market, there are a variety of diverse open-source, partly open-source, and closed data visualization tools with a variety of features available to extract and display all sorts of data.

In this chapter, a study carried out at the level of the available data visualization tools in addition to alternatives to the tools used by the company and that were used for the project, will be further be discussed and explained as to why they were used or not for this project.

The data visualization modules created for the different portals have the purpose to familiarize the user in working with both portals and as a final point help him in managing a merging of both portals. In addition to this, the creation of new previously inexistent modules inside the portals gives users new ways to study their trademark clients.

2.1 Existing work/Related tools

2.1.1 Web2py portal

Web2py [34] is one of the tools used to create and publish analytical modules for the user in a web2py portal.

Web2py is a Python programming language-based open-source online application framework, allowing web developers to use Python to create dynamic web content. Although a web developer may design a form from scratch, if necessary, Web2py is meant to assist decrease tiresome web development activities like building web forms from start to beginning.

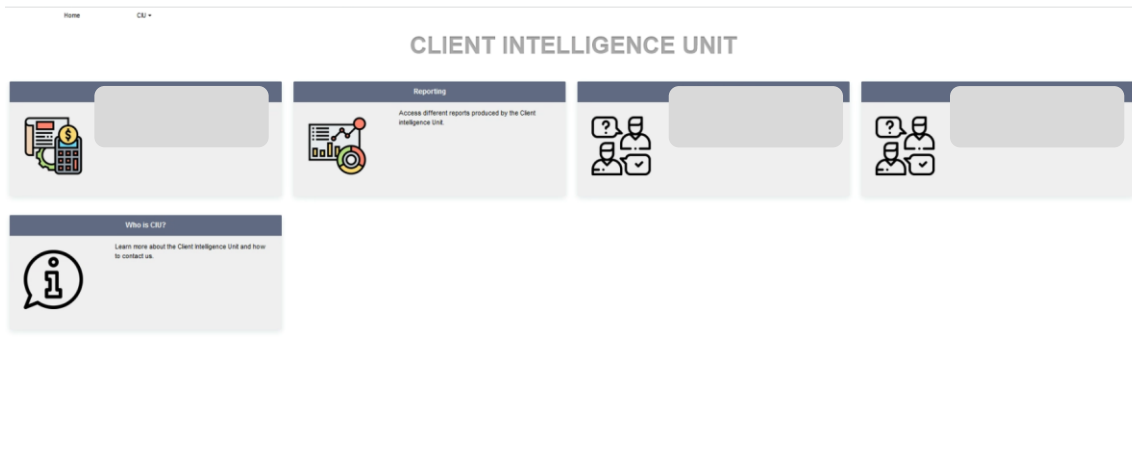


Figure 2.1 – Web2Py portal

Web2py was created as a teaching tool with the goal of being simple to use and deploy. As a result, there are no project-level configuration files. The Ruby on Rails and Django frameworks inspired the creation of web2py. Web2py focuses on quick development, using a model–view–controller (MVC) architectural pattern, and encourages convention over configuration.

Alternatives to this technology could have been Flask, CherryPy, Django among others, and are also written in Python. However, no other option other than web2py was taken into consideration since one of the portals to upgrade was already set up in web2py and running for quite some time, also it was already quite compact, containing many distinctive features and data analytical modules. The implementation of Web2Py was done before joining the department for the MI project since there was an urgent need for a free environment that could support the integration of python code in HTML [12]. Python was flexible and easy to create forms with. The membership handling is easy and finally, it was urgently needed, so Web2Py was chosen. The currently running Web2Py portal (Figure 2.1) deployed for the department has the possibility to generate reports and dashboards for different PSA and PSR needs.

2.1.2 Pentaho portal

The second technology used is Pentaho [16]. The Pentaho is also already in place with data analytical features. This tool is a Business Intelligence (BI) software that includes features like data integration, OLAP services, reporting, information dashboards, data mining, and extract, transform, and load (ETL). Pentaho is one of the most popular open-source BI solutions in the world, with a wide range of data integration, mining, dashboarding, customized ETL, and reporting capabilities. Hitachi Data Systems purchased Pentaho in 2015, and it is now part of Hitachi Vantara. Modern Pentaho Business Intelligence Solutions enable users to combine data from several sources, perform real-time analysis, and assist in presenting results in an aesthetically compelling

manner, therefore firming and improving decision-making processes at all levels of a business. Pentaho Business Analytics is available in two versions: a community edition and an enterprise edition.



Figure 2.2 – Pentaho portal

Pentaho is the portal used for deploying the report created in B.I.R.T. (Business Intelligence and Reporting Tools), for the second deliverable but also allows the creation of analytical modules directly. The B.I.R.T. will be explained in the next subsection of the thesis.

There were no alternatives taken into consideration besides Pentaho since the second portal running for the department is a Pentaho portal (Figure 2.2) that is up and has been running for quite some time and additionally hold reports and dashboards that the different departments use for data analytics.

2.1.3 B.I.R.T. tool

Eclipse B.I.R.T. [2] is a Java/Java EE-based open-source reporting system that interfaces with your Java/Java EE application to provide visually appealing results. The Eclipse Foundation sponsors B.I.R.T, an open-source technology platform that encompasses a visual report builder and a runtime component for Java and Java EE environments.

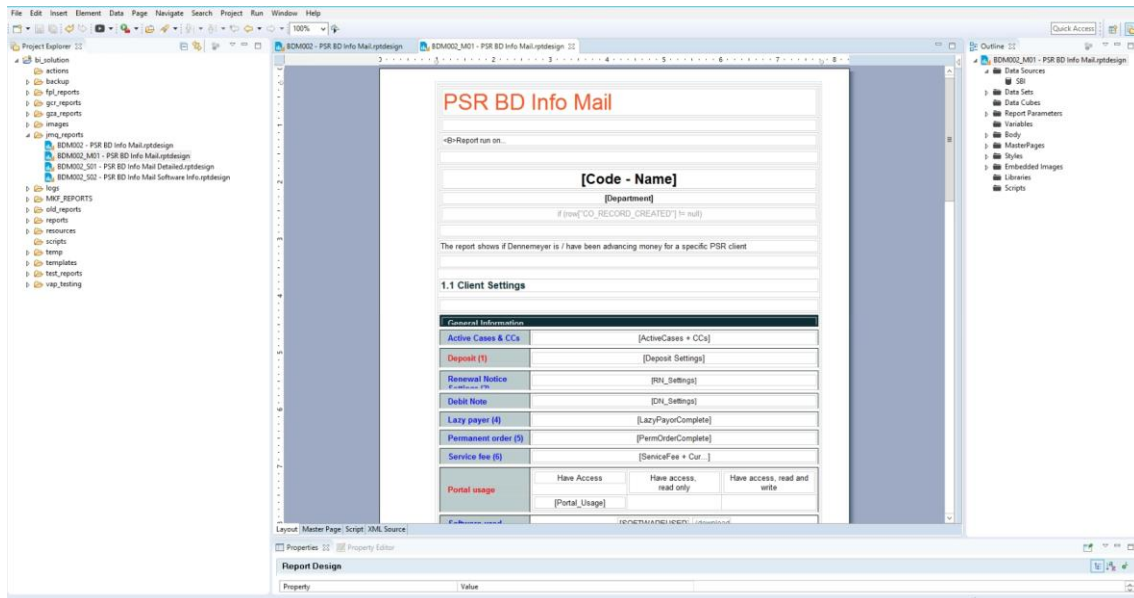


Figure 2.3 – B.I.R.T. tool

Reports, data visualizations, and charts may be built with these tools and then included in Web-based and rich client applications. Embedded BI solutions, such as the B.I.R.T., may be utilized to make real-time decisions as well as track and analyze previous data or ongoing trends. Lists, charts, crosstabs, forms and documents, and compound reports with many features are all examples of reports that may be built using BIRT.

To compile a report, this business intelligence and reporting tools support data gathering from a variety of sources, including databases, Web services, and Java objects. Data from several sources may be integrated into a single report. The information may be sorted and aggregated to generate conventional reports or adjusted to meet the needs of the end-user. Because the output may still contain too much data for decision-making, JavaScript may be used to add business-specific logic that enhances the utility of reports. From Excel spreadsheets to PowerPoint presentations to PDF documents, the generated data can be delivered in a range of visual and text formats.

This reporting tool (Figure 2.3) is used for the creation of the report on the trademark clients for the second deliverable and was already set up for usage with the Pentaho portal. The software works as simple as, creating the report in the B.I.R.T, saving it, exporting it, and importing it into Pentaho. As soon as you log in to Pentaho the user can use the created report if he has the necessary user rights. These user rights are defined for every user created and imported in Pentaho.

Alternatives for this tool could be any data visualization tool that allows report creation from external data sources, like Power BI. Since these also allow the creation of report-like visualizations that can be printed or saved in A4 format, and this in different file extensions, among many other features. But since a Pentaho portal was already up and running, no other tools were taken into consideration.

2.2 Other tools used

2.2.1 Microsoft SQL server

Microsoft SQL Server [26] [41] is a leading manufacturer in database technology [31]. It's a relational database management system that may be used for a variety of purposes, including business intelligence, transaction processing, and analytics.

Microsoft SQL Server, like other RDBMS software, is based on SQL, a standardized programming language used by database administrators (DBAs) and other IT professionals to administer databases and query the information they contain. Transact-SQL (T-SQL), a Microsoft SQL implementation that adds a set of proprietary programming extensions to the standard language, is bound to Microsoft SQL Server.

The department uses a Microsoft SQL Server to store all its information regarding trademarks, patents, clients among many others. This SQL Server is running on a windows server named *datalake*. The Pentaho portal and the Web2Py portal's analytical modules query the patent, trademark, and client information from this server.

2.2.2 SSMS

SQL Server Management Studio (SSMS) [27] provides a unified management environment for any SQL system. SSMS provides a set of tools to monitor, configure, and administer SQL Server and database instances. The tool can be connected to local or external databases and data warehouses can additionally be used to deploy, monitor, and upgrade the data-tier components that can be used by different applications.

SSMS (Figure 2.4) is a single, complete application that combines a wide range of graphical tools with a variety of powerful script editors to provide developers and database administrators of all skill levels access to SQL Server.

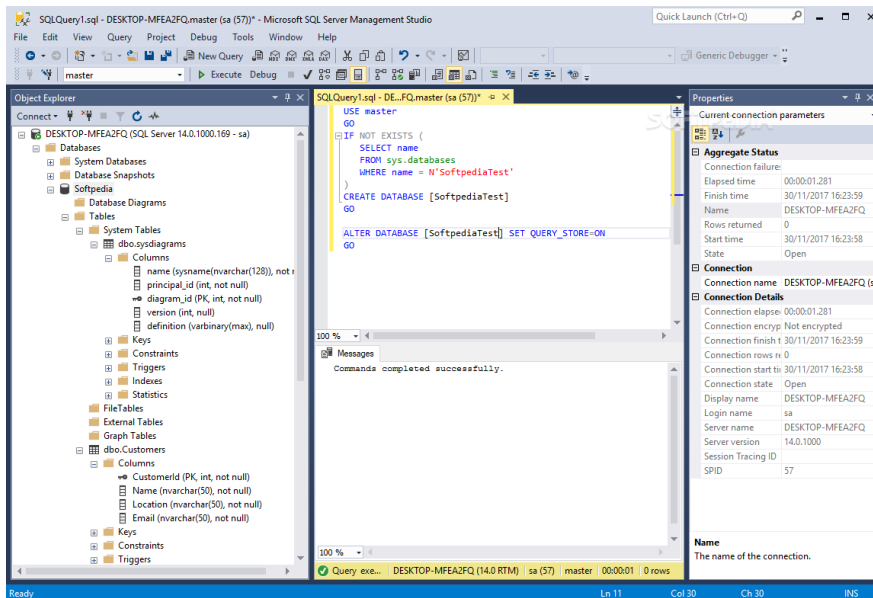


Figure 2.4 – SSMS example, extracted from [28]

The usage of this tool is crucial to the department since basically everyone uses it daily and most importantly, for the sake of this project, this tool was used for the query creation for the web2py chart and the Pentaho report. SSMS is the base program that enables us to ensure that we get exactly the information we need from the different databases. With the help of SSMS, the department can connect to the Microsoft SQL Server running on the *datalake* server and query any information from there.

2.2.3 Jupyter Notebook

Another tool used for the initial Python code testing with a matplotlib [11] [15] chart is Jupyter Notebook [17] [35].

This is a web-based server-client tool for editing and executing notebook documents. The Jupyter Notebook App can be run locally on a computer without internet access remotely on a server and accessible through the internet. These notebooks are human-readable, and the code can directly be executed with the according result shown in the output appearing underneath the code (Figure 2.5). This is a very interesting tool used tool that in this case quickly and efficiently allows Python code testing.

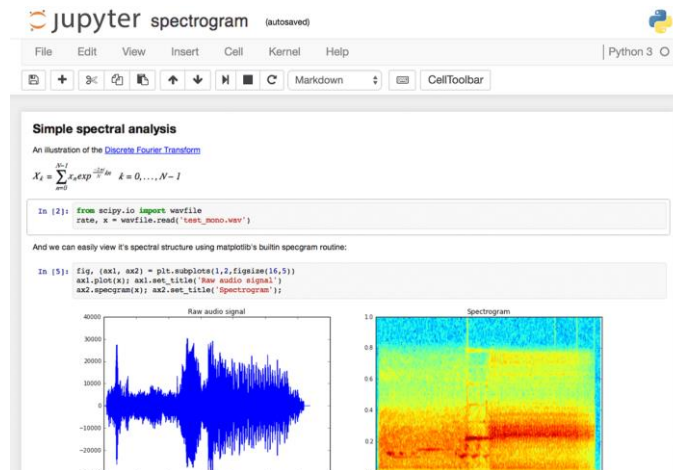


Figure 2.5 – Jupyter Notebook example, extracted from [5]

This tool has become a very popular tool among data scientists due to being open-source, flexible, interactive, and very easy to use. In fact, data scientists tend to like the tool for practicing and sharing various codes, quick prototyping, and exploratory analysis [5] [13].

2.3 Pipeline and deployment management

2.3.1 Git

Git [8] is the version control tool used by the department to track and push changes and new developments into several projects, the web2py portal being one of them. Git makes collaboration easier, by allowing many people to work on the same project. The changes are tracked via named branches and if needed a project can be reverted to a previous version of the project from an earlier time. Finally, the changes will all be merged into one main source. Consequently, with Git you can find out what exactly was changed in a project, who did the change and why through a small description that gets added by a user Git tracking its changes.

2.3.2 Azure board and project management

The department also uses the azure board [37] (Figure 2.6), which is an interface that helps the team track tasks, bugs, and features within a software project. This tool helps teams to maintain track of all of your ideas at every level of development so that your team is aware of all of the essential code modifications related to your work procedures.

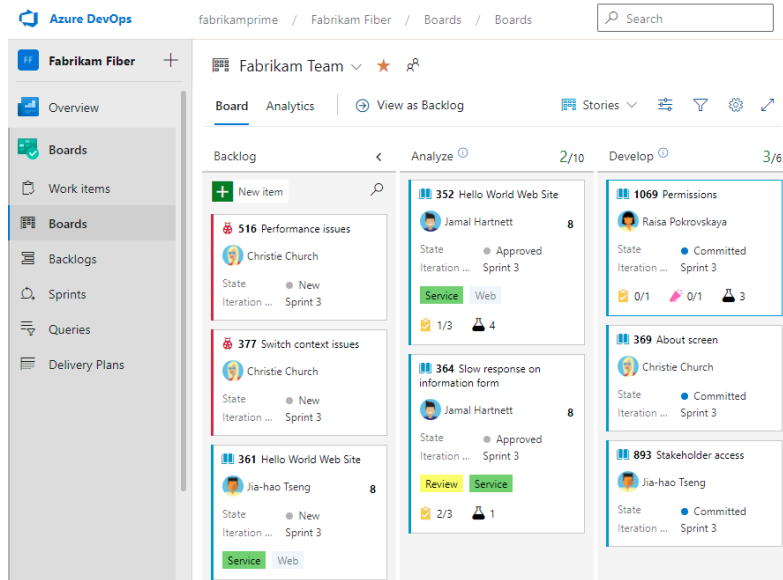


Figure 2.6 – Azure board example, extracted from [22]

The team in the department also works with agile methodology [38] sprints [25], which consists of the condensed period during which a scrum team works to finish a specific amount of work. Scrum and agile approaches are built around sprints and getting sprints right can help the team ship better products with fewer issues.

The Agile (Figure 2.7) approach is a style of project management that divides a project into stages. It necessitates ongoing engagement with stakeholders as well as continual development at each level. Teams cycle through a process of planning, executing, and assessing once the job begins. Collaboration is essential among team members as well as project stakeholders.

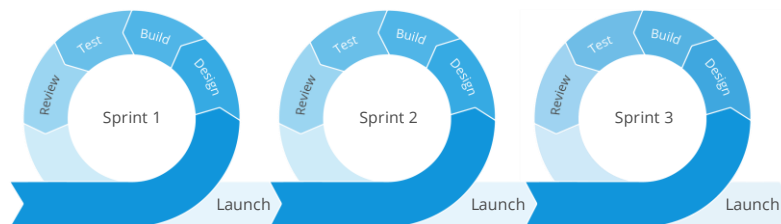


Figure 2.7 – Agile methodology sprints, extracted from [43]

In addition to the board being used for tasks, bugs, and features and the agile methodology sprints for project management, the CIU team also started to slowly implement the CI/CD [39] development method.

CI and CD are the abbreviations for Continuous integration and continuous delivery/continuous deployment. In its most basic form, continuous integration (CI) is a modern software development method that involves making incremental code changes regularly and consistently. CI-triggered automated build-and-test stages guarantee that code changes merged into the repository are trustworthy. The code is subsequently given

quickly and accurately as part of the CD process. In the software sector, the CI/CD pipeline refers to the automation that allows incremental code changes to be delivered swiftly and reliably to production from developers' workstations.

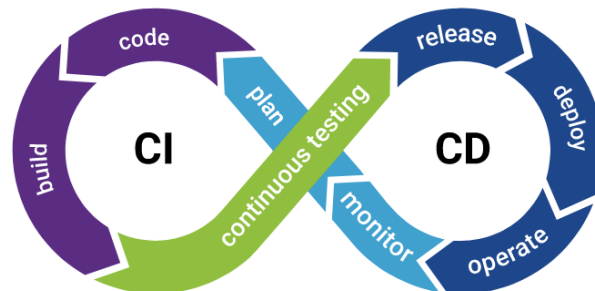


Figure 2.8 – CI/CD, extracted from [39]

Organizations may use CI/CD to ship software fast and effectively. CI/CD enables an efficient procedure for bringing products to market quicker than ever before, continually sending code into production, and assuring a continuous flow of new features and bug fixes via the most efficient delivery mechanism.

2.4 Conclusion

To conclude, all the previously mentioned tools and deployment management methods were already in use by the department before I joined the project, consequently these tools were also used for the project; Web2Py and Pentaho for portal hosting, SQL server for the database, Git and DevOps for version control and repository and finally agile as a deployment methodology. However, for the Python development I could choose whatever IDE I wanted and decided to go for Jupyter Notebook, because it offered an easy way to execute and test the code without having to execute whole blocks of codes. Additionally, the fact that it displays the results right after the lines of codes facilitates greatly the debugging and testing.

Chapter 3:

Analysis and planning

Before beginning analyzing the existing solutions, stakeholders were defined. The stakeholders for the project were the clients (users from the company), the company supervisor, managing members of the trademark department and the CIU department. Since the tools are all exclusively for internal usage, the company supervisor played the role of the clients, being a user testing the developed modules and validating them, which in reality are people from the department that use the analytical tools. Additionally, the trademark department, the CIU department and the company supervisor also took part on the testing process of the modules.

The approval for the developed modules was given exclusively by the project supervisor.

3.1 Deliverable 1 – Late instruction counter

The first module involves the collected data on late instructions we get. Late instructions are payment instructions the client sends for the renewal of trademarks after a fixed date that the client has to instruct us. This being an infringement in the process of trademark renewals, makes us fine them with a late instruction fee, if they want us to continue with the renewal of the trademark. Getting the late instruction information for specific clients, to see how their paying behavior is, was done in a quite complicated and inefficient way. If someone in the department requested the CIU to provide the number and amount of late instructions a specific client has for a certain date range, the CIU team had always to query this information out of the database, export it to a pdf or excel file and send it to the user. This was done very inefficiently since this would lead to a massive amount of similar incoming requests mostly from the trademark department, in case they wanted to have an overview on the client behavior related to late instructions.

To solve this issue, the creation of a first chart was decided upon, in addition to the technology that would be used for this chart. Since the technology selected was web2py, the analytical module brings an upgrade of the existing web2py portal with a new analytical tool in place and solves the issue of people from the trademarks department asking for client information about late instructions. However, this wasn't the main reason for the creation of the report but gaining further knowledge on how the web2py portal works.

3.2 Deliverable 2 – TM Client report

The fact that the CIU department portals provide very few analytical features for trademarks, has consequently not much client analyzing that can be done. There is no quick and efficient way to find out the most important information about specific clients by using a few clicks. If the trademark department wants to get a summary of the most relevant and characteristic information about a client, it needs to contact the CIU team about what information they would want to see for which client, and after querying the desired records for a specific client, they would get a pdf file or an excel file with the required information about the client in table format.

Finding a solution to this issue is of course very interesting for the trademark department, however, this new report was also created with the purpose of gaining some knowledge on how the Pentaho portal works.

3.3 Analysis of the problem (Merging of portals)

3.3.1 Fishbone diagram – Problem

For the analysis of the problem, a more extended fishbone diagram [36] was created [7] with the causes and consequences of the problem. Starting from the main issue, the consequences and causes were divided into six categories, people, machine, measurement, material, method, and environmental like in Figure 3.1.

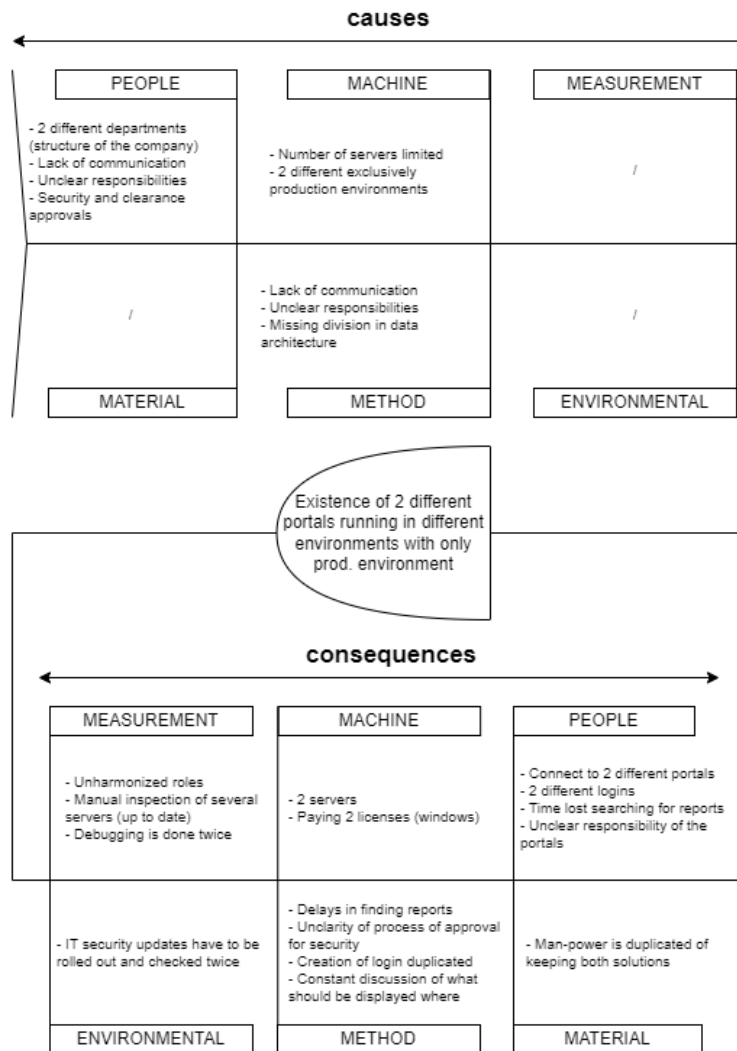


Figure 3.1 – Fishbone diagram: causes & consequences

3.3.2 Workflow of portal development – Login creation

For the login creation for users wanting to use the portals, the swimlane diagram [30web2py] looks like in Figure 3.2. All the upcoming workflows will contain rectangles representing actions, diamonds representing a condition, circles representing the end and the start of the workflow or a specific ending by doing an action and finally some workflows contain parallelograms that mean the execution of another workflow. The main issue, in this case, is the constant asking of access to the CIU team for access to one or both portals. This goes over a form that is sent via email and needs to be filled out for the CIU team to know, to which portal the user wants to access.

In addition to this, the CIU team must create the credentials for the users manually for one or both portals, inside the used user’s database that is linked to the specific portal.

The login creation for the Web2Py portal is done by using a Web2Py table named auth_user, adding the new user directly into the table. For the login creation of a Pentaho user, an external SQL table named Pentaho_users, is used, creating a new record in there

for the new user. For both login creations, the according role is also added, since not every user can access every module available in both portals.

After the creation of the login has been done, the user will be notified via email about his username, the random password that was generated with the login creation, and the links to access the required portals. Lastly, it is included in the email that the user should change his password to a new personal password as soon as possible.

In addition to this, the passwords in Web2Py do not expire, and therewith comes the security issue of having the same password for a very long period until the user decides to change it, which could take years.

This Login creation workflow (Figure 3.2) is defined by the interaction between the user, the CIU team and the CIU members responsible for the Web2Py db and Pentaho db. The lines show the interaction between the different stakeholders in this Process.

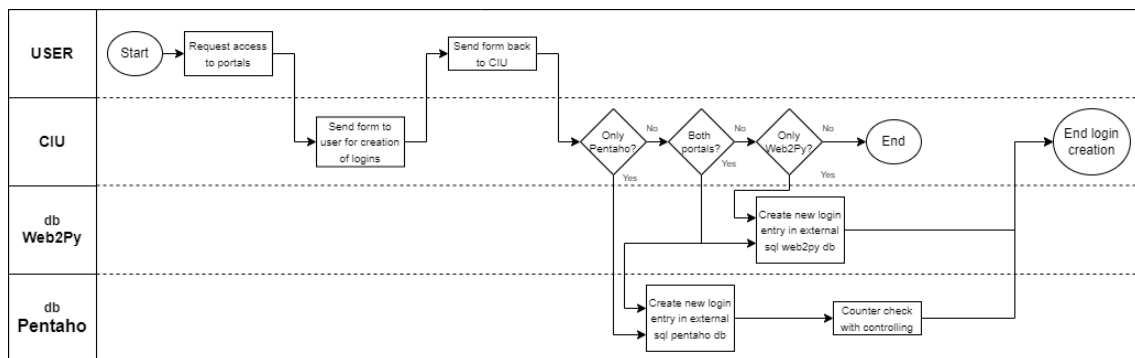


Figure 3.2 – Login creation (old - Swimlane diagram)

3.3.3 Workflow of portal development – Creation of a report

The creation of a report swimlane diagram looks like in Figure 3.3. Any department in the company can come up with a report idea that would be beneficial to the analytics used in the company. The report creation procedure, even though it is one that cannot be altered or enhanced with this project, it is still interesting to know how it is handled.

The CIU department has a general email for inquiries. Upon receiving a request, the team decides whether the new analytical module idea is interesting to implement and brings value to the company.

After discussing and analyzing the idea, upon the acceptance of the request, the technology used to implement the module is discussed and the module is developed.

Upon successful testing and validation of the module, it is put into the production server, and the user who requested the module is notified via email.

The creation of a report workflow (Figure 3.3) is defined by the interaction between the user, the CIU team and the CIU members responsible for the Web2Py development and Pentaho development.

Even though this process has been studied, the report creation workflow was not upgraded since there will still be two portals running.

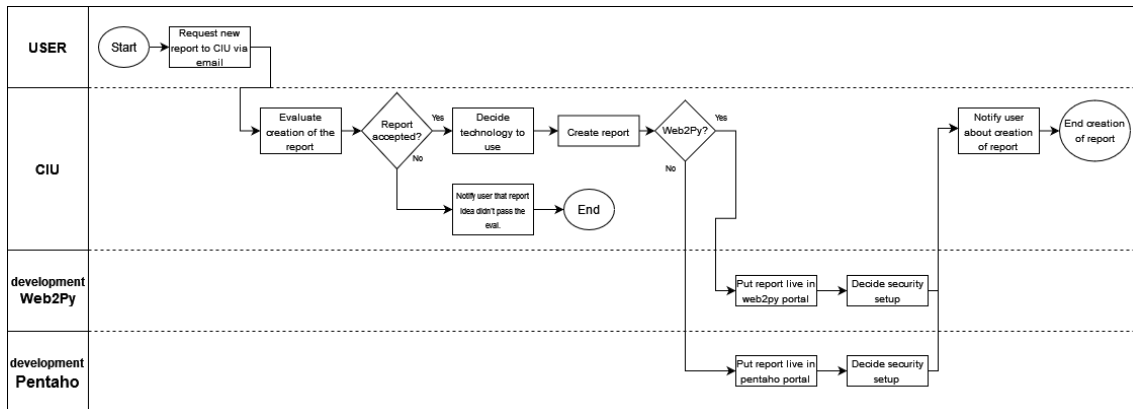


Figure 3.3 – Report creation (Swimlane diagram)

3.3.4 Workflow of portal usage – Generation of a report

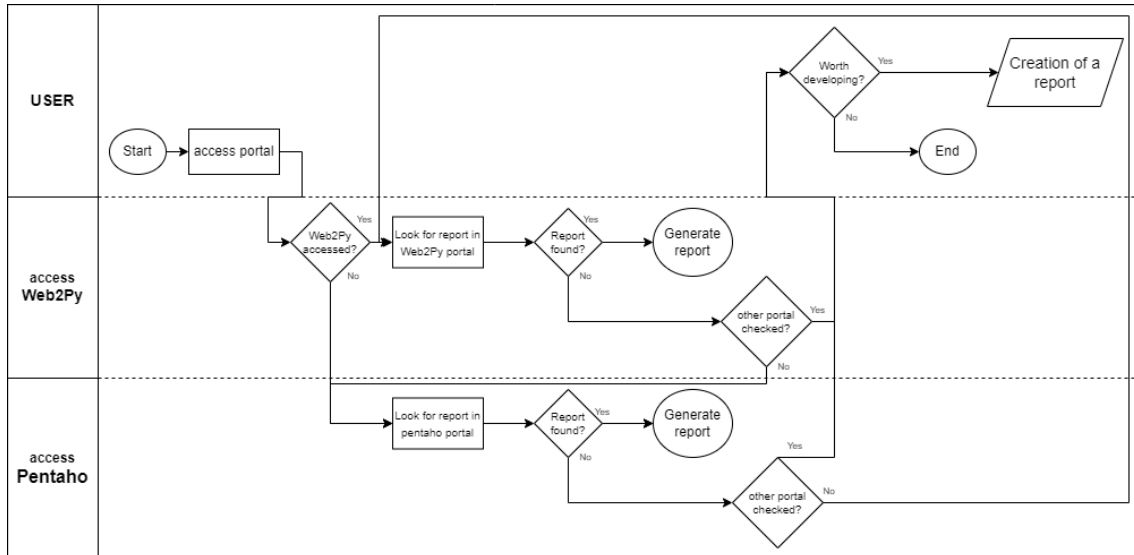


Figure 3.4 – Report generation (old - Swimlane diagram)

The generation of a report process is presented in Figure 3.4. Since there is a web2py and a Pentaho portal available and running on a production server. In the case of this workflow (Figure 3.4), a user who does not know where to find a specific report needs to check both portals, consequently in the worst case, login twice into both portals. After finding a specific report, it can be generated. However, if a specific report cannot be found in any of these portals, and the analytical feature idea is a good one, it can be sent as a request to the CIU team, for them to determine if the analytical feature is worth developing, jumping into the workflow for the creation of a report (Figure 3.3).

The creation of a report workflow (Figure 3.4) is displayed by showing the interaction of the user with the two different portals named “access Web2Py” and “access Pentaho” for the generation of a report that may already exist.

3.3.5 User role selection

The user creation for the Web2Py portal was done by adding the new user manually into the internal Web2Py database and consequently when a user tries to login into the Web2Py portal, the internal Web2Py table `auth_users` was queried in order to enable someone to log in, just like in Figure 3.5.

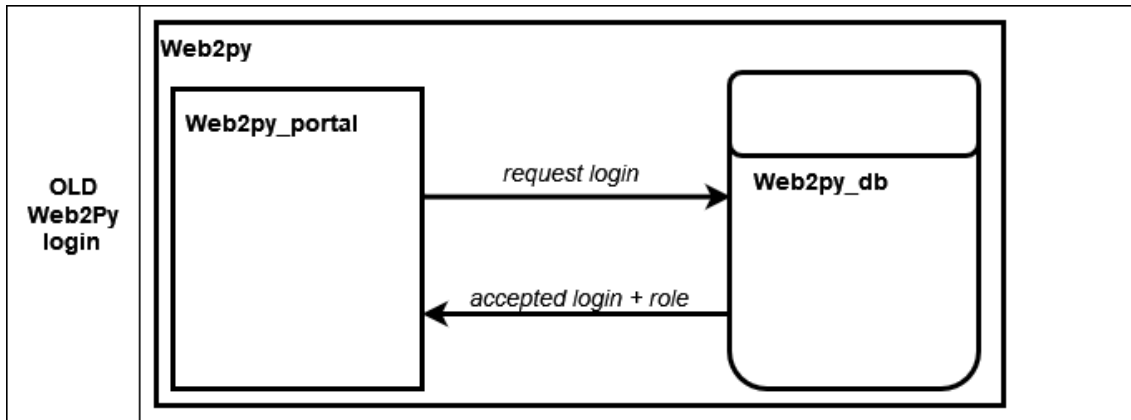


Figure 3.5 – User role query Web2Py (old)

In Figure 3.6 you can see how the login into the Pentaho portal is done. “OLD/NEW” means that the way the login was done was kept the same.

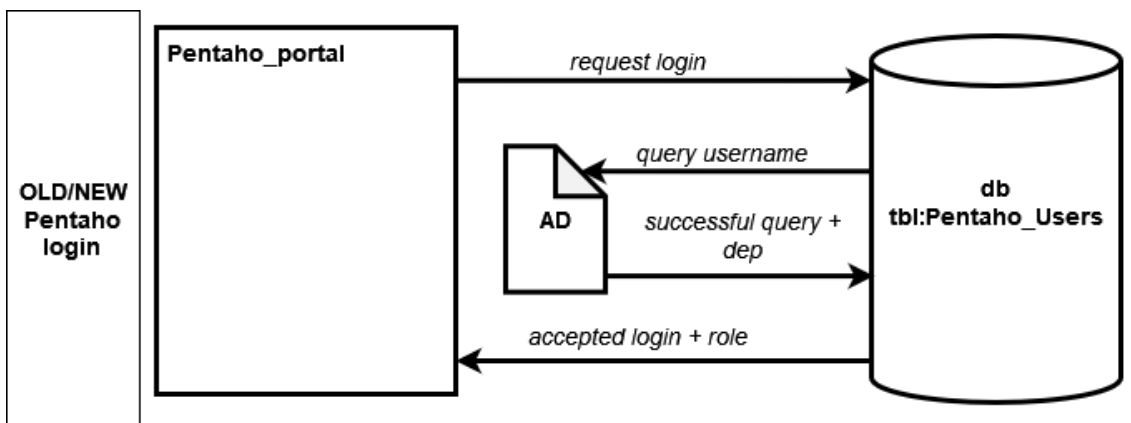


Figure 3.6 – User role query Pentaho (old & new)

The user creation for the Pentaho portal was done similarly to the one for the Web2Py portal. A new user with the appropriate role was added to the external Pentaho table named `Pentaho_users`. For the user selection, however, the login is requested by first querying the submitted user email address from the `Pentaho_users` table, and if the record exists in there, it is additionally queried from the AD to compare if the password entered also matches the one the user has inside the AD.

Moreover, two production windows servers were running, both with one portal each. No development server, no staging server but only 2 production servers whereas Pentaho portal was running with tomcat and Web2Py with Apache in Xamp.

3.4 Proposing a new solution

3.4.1 Deliverable 1 – Late instruction chart

After analyzing deliverable 1 and the need for a chart that shows the amount of late instructions for a specific client, a simple draft was designed on how the module should look like. The draft decided upon is like in Figure 3.7, where we select first the client and the date range for which late instructions should be looked for. After selecting these two fields, the corresponding chart and table will be generated.

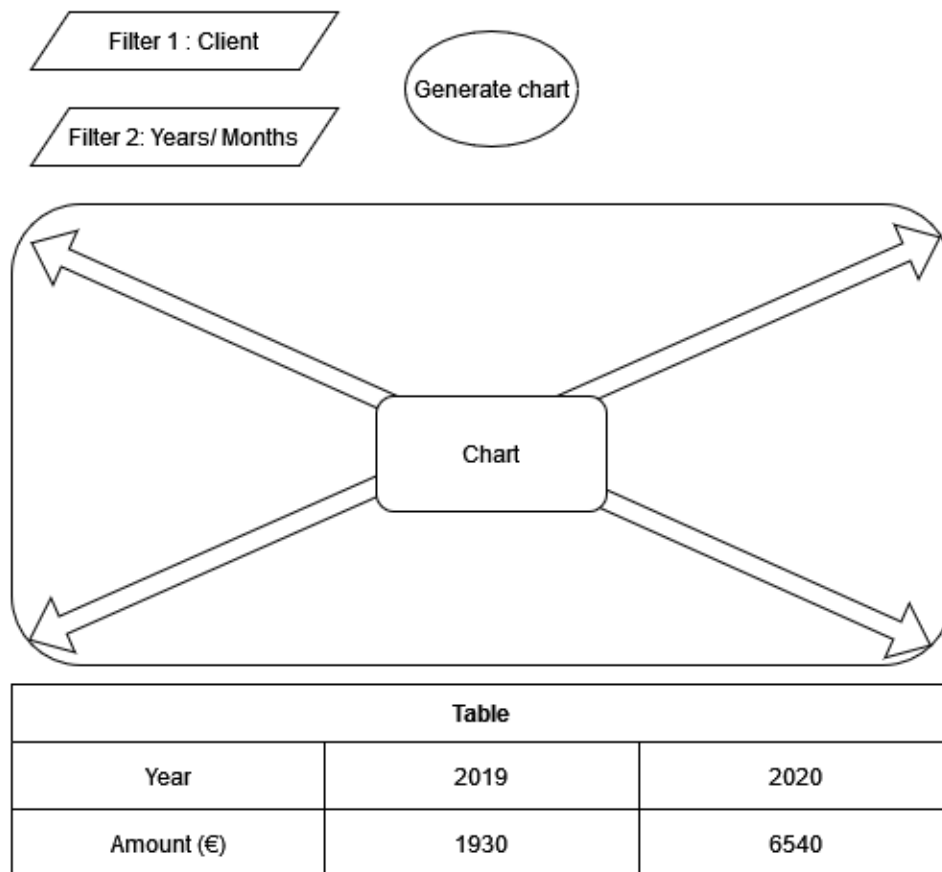


Figure 3.7 – Late instr. chart draft

3.4.2 Deliverable 2 – TM Client report

After analyzing deliverable 2, a client report design was drafted just as in Figure 3.8, where the main characteristics are the report name at the top, followed by the client's name, and underneath a table-like view for the most important client information.

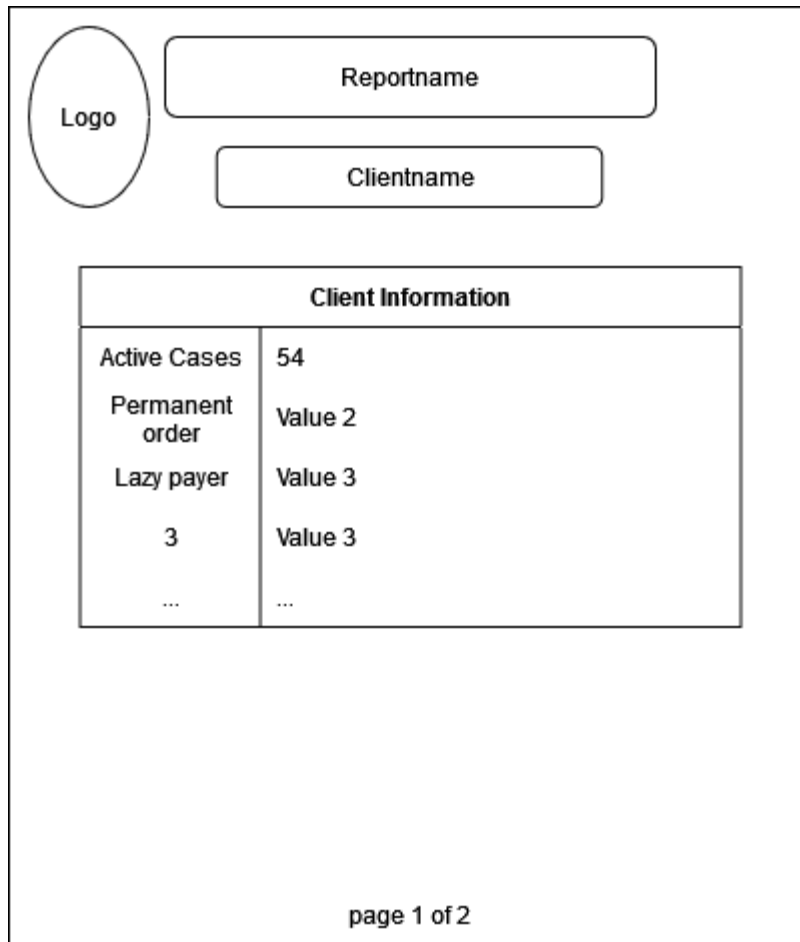


Figure 3.8 – TM client report draft

3.4.3 Merging of portals

For merging the portals, there were 3 options studied. One solution could have been including all the features from the Pentaho portal into the web2py portal or the other way around. However, recreating complete modules would take too much time since these modules are not just simple dashboards but more complex projects. This idea was quickly put aside, also since Pentaho and Web2Py work with different technologies, and importing the content from one portal to the other would require a deep knowledge with both portals.

The other option was to continue using both portals. While implementing the web2py in an iframe inside a tab in the Pentaho portal just like the draft in Figure 3.9.



Figure 3.9 – Merged portals draft

This design and idea were kept since Pentaho has the login already connected to the AD, it was decided to get rid of the web2py login by integrating the Web2Py inside the Pentaho portal, still giving the user the sensation as if it was only one portal.

For the single login to be able to work for both portals, the solution is to log in once over the main portal that will be the Pentaho portal. Now the username is saved, and when clicking on the CIU tab of the Pentaho portal, the username is passed to the iframe with the link to access the Web2Py portal and automatically logged in, so that it looks like the user is not switching portals.

3.4.4 Workflow of portal development – Login creation(new)

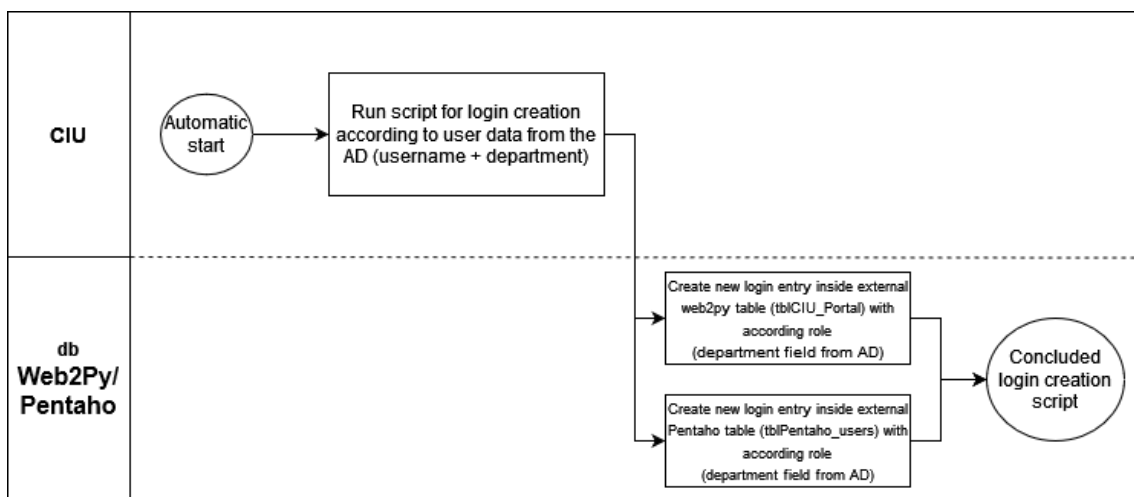


Figure 3.10 – Login creation (new - Swimlane diagram)

Since there will be a new solution for merging the portals, a new solution for the user login creation was defined. For the new automatized solution, the department will be

automatically run a script every Monday to query the AD and add new users to the according Web2Py and Pentaho user tables with their specific default roles.

3.4.5 Workflow of portal usage – Generation of a report(new)

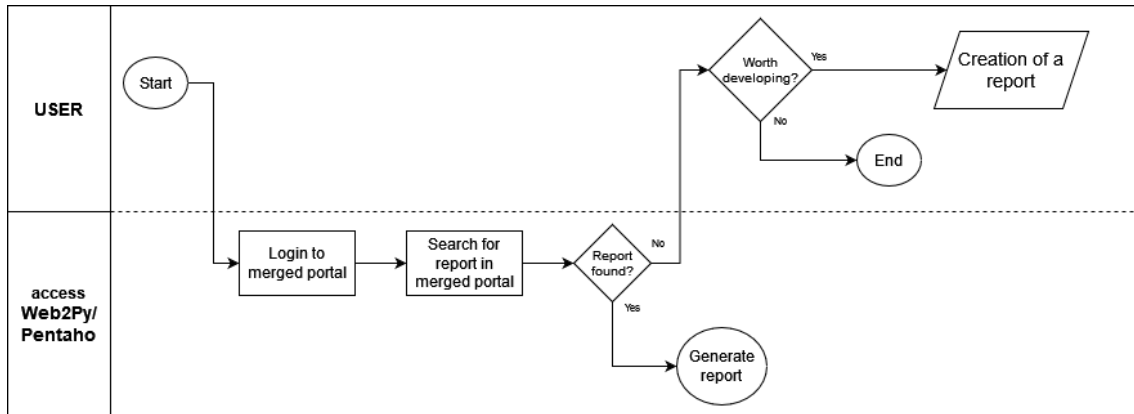


Figure 3.11 – Login creation (new - Swimlane diagram)

The new solution for the generation of a report is, nearly equal to the initially analyzed. The main difference for the new solution is that the users don't have to log in twice to be able to access all the features available in both portals since they are merged. Finally, if the user does not find a report, mainly because there is none for that specific purpose, a request has to be created for a new report.

3.4.6 User role architecture

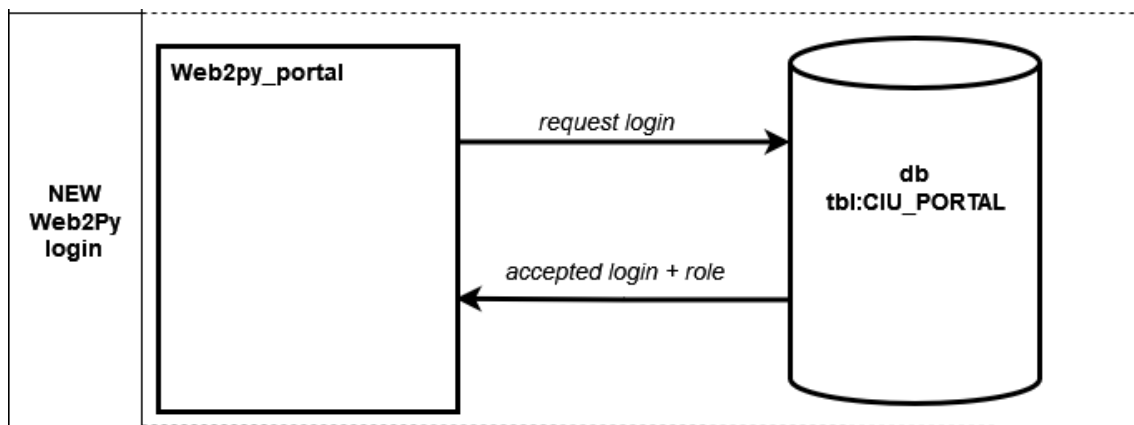


Figure 3.12 – User role query (new)

The CIU team wanted to have all the needed user information inside an external database. For the new solution, when the automatic login of a user into the Web2Py portal over the Pentaho portal happens, an external users table named CIU_Portal is queried for login instead of the internal Web2Py table auth_user.

3.5 Study on data visualization tools

3.5.1 Power BI

The first tool that was studied was Power BI [6] [42]. Power BI is currently a very used data visualization [32] tool that is on the market. In addition to being very intuitive and having an appealing GUI (Figure 3.13), the functioning of the tool is easy to learn and use [3].

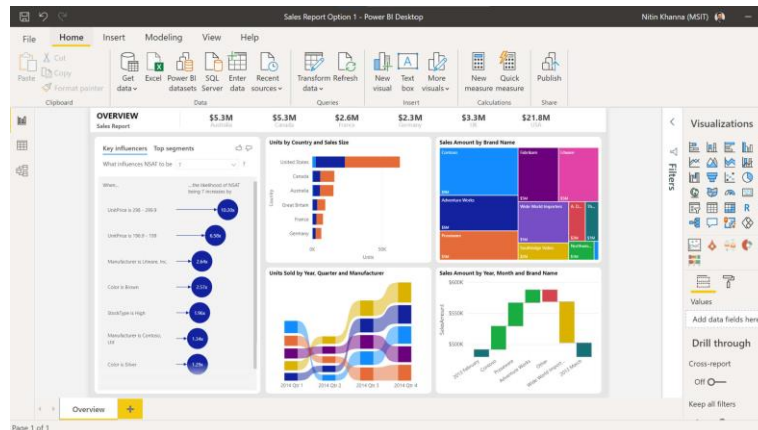


Figure 3.13 – Power BI example, extracted from [6]

Power BI is a set of software services, applications, and connections that work together to transform different data sources into logical, visually engaging, and interactive insights, whereas data could be stored in excel spreadsheets, cloud based or on-premises data warehouses. There are currently 3 Power BI per-user licenses available: free, pro and premium per user license. Currently the company has Microsoft 365 E5 licenses for every user inside the company. Many used Microsoft applications are included in this license, one of them being Power BI Pro.

The analysis of this tool was tackled by going through the features listed in Table 3.1 and trying them out with the data analytical tool.

Power BI was tested in more detail than the upcoming two tools, since the department was already leaning more towards it, once there are already licenses for this tool.

3.5.2 Alteryx

The second data visualization tool is Alteryx [23], also a very known Software solution that allows users to access, alter, analyze, and export data efficiently. This can be done on a one-time basis, or it can be done on a recurrent (daily, weekly, or monthly) basis (called a "workflow"). Alteryx constructs workflows (Figure 3.14) in such a manner that they become a type of process documentation, allowing users to examine, contribute on, support, and improve the workflow. It is also possible to read and write data to databases,

files and APIs. In addition to this, Alteryx also has capabilities for predictive analytics and geospatial analysis.

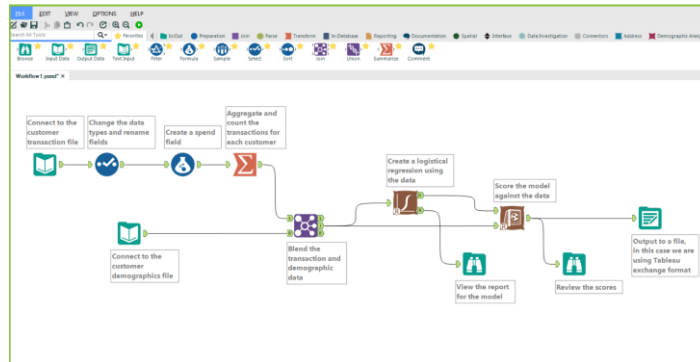


Figure 3.14 – Alteryx example, extracted from [24]

Alteryx works with workflows by drag & dropping different process nodes (Figure 3.14) that allow data filtering, transforming data, showing intermediate data, and visualize specific data, among many others.

Alteryx allows the automation of repetitive tasks by creating the specific workflow in its simple but powerful GUI. The tool is also able to connect to many different data sources.

The GUI is quite clean and creating visualizations is very easy and straightforward. Process nodes are used and connected among them, each having their own role in the workflow.

3.5.3 Qlik

The final data visualization tool studied, was Qlik [20]. The tool offers many user-friendly features, providing advanced data analysis and powerful visualization services like shown in Figure 3.15.



Figure 3.15 – Qlik example, extracted from [21]

Qlik sense allows the user to associate data from different sources, just like the previous tools. The tool serves as a data analytics platform for a wide range of users, being able to handle multiple different visualizations. However, the tool is more to show and analyze data in nice graphics, whereas QlikView allows data manipulation through scripting. Qlik sense allows the creation of visualizations, charts, dashboards and analytics app that can be use online and offline.

3.5.4 Analysis of the tools

A small report was created while studying the three above mentioned data visualization tools.

The most important part of this report to be checked was the table (Table 3.1) containing a set of important features a new data visualization tool should contain to be of use within the department. In addition to this since the department was leaning more towards Power BI, additional meetings were held where the main needed features were recreated with Power BI and shown to the CIU team for approval. Consequently, Power BI was analyzed in more detail than the other two tools.

In Table 3.1 we have the following symbols:

- “✓”, “✖”: The tool supports or does not support the feature.
- “✓✖”: The tool supports the feature but difficultly.
- “(✓)”: According to the research done, it should support the feature.
- “(✖)”: According to the research done, it should not support the feature.

Functionalities	Power BI	Alteryx	Qlik
Multiple access?	✓✖	✓	✓
Security management?	✓	✓	✓
Write back into db	✓✖	✓	(✖)
Simple visualizations	✓	✓	(✖)
Easy to use/learn	✓	✓	✓
Big & active community	✓	✓	✓
Support included	✓	✓	✓
Frequent updates	✓	✓	(✓)
Works with Python	✓	✓	✓✖
Works with JS	✓	✓	✓✖
Works with HTML	✓	✓	✓
Drag & drop visuals	✓	✓	✓
Handles automatic sendings	✓	✓	✖
Connection to birt	✖	✖	✖
Data science models?	✓	✓	✓
Embedding reports into portals	✓	✓	✓
Version control	✓	✓✖	✓✖
Already paying for a license	✓	✖	✖
Pricing	€	€€€	€€

Table 3.1 – Defined features required for the data visualization tools

Since the company was already paying for a Power Bi pro license, the “pricing” in the table was accordingly set to the lowest.

According to Table 3.1, even though Alteryx seems to be the best option, Power BI was more convincing mainly due to the pricing aspect also due to the already acquired knowledge inside the company.

The report about the visualization tools is an asset to the company because it contains a detailed study on three robust data visualization tools. This study enters the aspect of pricing, compatibility with already used programming languages, support of external data sources, among many other things features, and may allow the team to decide on the solution to choose for a further improvement of the portals.

Finally, the three tools were studied by trying to recreate analytical features that were already available in both portals and present them during meeting sessions. This was done to see the extend of the tool and to see if it would be an adequate replacement for the existing modules. However, since Power BI was studied in more detail, these analytical module recreations in the studied data visualization tools were done by putting more effort into the Power BI study.

3.6 Summary and conclusion

To summarize, this chapter contains an analysis of the previous solutions that were in place in addition to the new solutions that improved the old ones drastically.

Previously to check for late instructions, users needed to ask the CIU to query that specific information and they didn't have a visual overview of the client's behavior related to the late instructions. The first analytical module allows the user to directly search for a client's late instructions information without any issue and having a clear image on the client's behavior.

The trademark department was also lacking related to reports available for trademark clients. The second analytical report allows them to quickly get a picture of the client by having his most important information quickly displayed in a clear way.

The solution for the first two analytical modules removes a burden from the CIU departments back, by allowing the users to look for specific information by themselves.

Jumping to the third deliverable which was the implementation of the merging of the portals, we have the analysis flowcharts on how the process looked like and, in the solution, how the new solution of the processes look like. The solution for the creation of new users lessens the burden of the CIU department by simply running a weekly script that queries the active directory and creates the usernames for new users into the Pentaho and Web2Py tables. Additionally, the way a user searches for a report inside the portals

has been improved. A user only needs to login once, and only needs to recall one password to access the portals. The automatic login to Web2Py from the Pentaho portal, allows the user to have the experience of only navigating through one portal.

Finally, three data visualization tools were analyzed and tested by recreating internally existing analytical features. Even though there was research made for the three tools, Power BI was the one more tested in detail. The analysis of the tools allowed me to determine if the tools could be adequate for a future usage inside the department, replacing the portals, or their visualizations.

Table 3.1 shows if the tools support the necessary features to be implemented and to be of use to the department and gives a clear insight on how they differ from one another.

After thorough analysis, the conclusion was that Power BI offers much more functionalities for a much more modest price. Additionally, the integration with the other systems also Microsoft based, is much easier and helps to have a much more coherent and sustainable environment.

Chapter 4:

Implementation

4.1 Development process

4.1.1 Deliverable 1 – Late instruction chart

Before I actually started, a deep knowledge of Python and Web2py had to be acquired in order to be more familiar with the technologies that were supposed to be used for the first deliverable. To get the said knowledge, the internship offered me the time to undertake a Python and a Web2Py Udemy course.

Now, Python had to be installed in order to run Python code. This was done by installing Anaconda [1], which is an open-source Python distribution.

Using the command line of a version controller, the matplotlib [18] library had to be installed by typing:

```
>> python -m pip install -U matplotlib
```

Which was then imported into our Python code or our Jupyter Notebook for usage.

Next, I started creating the Python code in my virtual machine, where I could freely develop without endangering the integrity or stability of the live systems. Jupyter Notebook was installed in order to be able to quickly develop and test Python code. For the cause of creating a module similar to Figure 3.7, the matplotlib library was used since the community using this library for visualizations is quite high and there can be found many questions and answers related to its implementation.

A connection [9] to the SQL database was also needed, which was established by using a predefined connection class. This function was created by a colleague of the department, to avoid conflicts when creating new modules and using different connection methods. Consequently, anyone who wants to connect to the database, uses the same connection class, so that the connection for all the analytical modules created in the Web2Py portal is done the same way, removing any conflicts on future adaptations of the code.

As previously mentioned, in order to query the client name, the connection class is used and run in the controller part of Web2Py by only defining the query we need, similar to the code in Figure 4.1.

```

From connection import Connection // import connection class
Connect = Connection() //create connection instance
clientList = connect.select_query("""SELECT DISTINCT COMPANY_NAME FROM
DATABASE""") //using the select_query function from the connection class

```

Figure 4.1 – Connection class usage

The populated list with the client names was then added to the first created dropdown menu that can be seen in Figure 4.1. The date filter was simply populated depending on the selection of the YEARS button which would populate year ranges like "2001-2010" or "2011-2020" in the dropdown filter underneath making the generation of the chart have the years shown in the x-axis or the MONTHS button, that would populate the filter dropdown menu with the different years and the generated chart would have the months of that specific year on the x-axis.

After developing the Python code to recreate the design of Figure 3.7 in Jupyter Notebook, it was run with correct data from the SQL database to display the module in Figure 4.2. A clean chart with the late instruction information of a specific client.

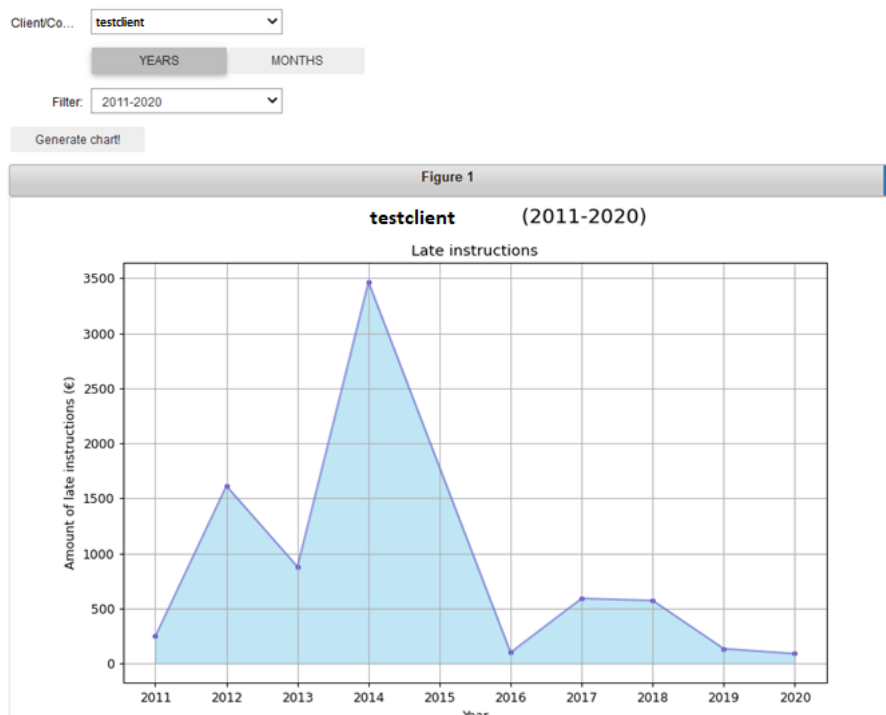


Figure 4.2 – Jupyter Notebook live Code - Late instr. chart

However, this module needs to run in the Web2Py portal of the department. For this purpose, a local version of Web2Py was downloaded from their official website and

installed on my virtual machine. Now having Python and Web2Py installed, the default Web2Py that was downloaded could already run locally without any issues.

Ultimately, we will need to run and test the module on the Web2Py portal that is on the production server, in other words, a new modified application that is running in Web2Py. This was done by cloning the portal application running in Web2Py of the production server and integrating that application in my local Web2Py portal.

Afterwards, version control also needed to be applied. Thankfully during the internship, I had access to their complete process documentation where I could follow “how to git track the Web2Py portal” in order to link my local version of the portal with the version running on the production server, to be able to push and track my changes into the development server by using Git.

Finally, we want to make the created module available in the local Web2Py portal application. For this purpose, a new view inside the “views” folder, named “LateInstructionCounter.html” has been created inside the local portal application. Additionally, a “late_instructions_counter.py” file has been created inside the “controllers” folder and a new card (Figure 4.4) has been created inside the existing PSR_reprting.html file, with the code displayed in Figure 4.3 that references the previously defined “LateInstructionCounter.html” file, whereas the “late_instructions_counter.py” main function is named the same as the newly created view.

Additionally, the design for the card had to be modified accordingly using CSS[4] inside the views or by creating an external CSS file.

```
<div class="col-xs-3 col-sm-3 col-md-3 col-lg-3 image-text card_outer" style="max-width: 25%">
  <a href="{=URL('late_instructions_counter', 'LateInstructionCounter')}"}>
    <div class="card border-primary mb-3" style="max-width: 100%;height:100%">
      <div class="card-header" style="text-align: center; ">
        <font size="4" color="white">Late Instructions Counter</font>
      </div>
      <div class="serviceImage">
        <div class="row">
          <div class="col-xs-3 col-sm-3 col-md-3 col-lg-3" style="margin:5%" >
            {=A(IMG(src=URL('static/images', 'LateInstructionCounter.png'), _width="100%"),
              href=URL('late_instructions_counter', 'LateInstructionCounter'))}
          </div>
          <div class="col-xs-7 col-sm-7 col-md-7 col-lg-7" style="margin:2%" >
            <a href="{=URL('late_instructions_counter', 'LateInstructionCounter')}"}>
              <font color="black">
                <p>Here you can look into more detail how lazy the clients are and count the late instructions.</p>
              </font>
            </a>
          </div>
        </div>
      </div>
    </div>
  </a>
</div>
```

Figure 4.3 – HTML code for new Card

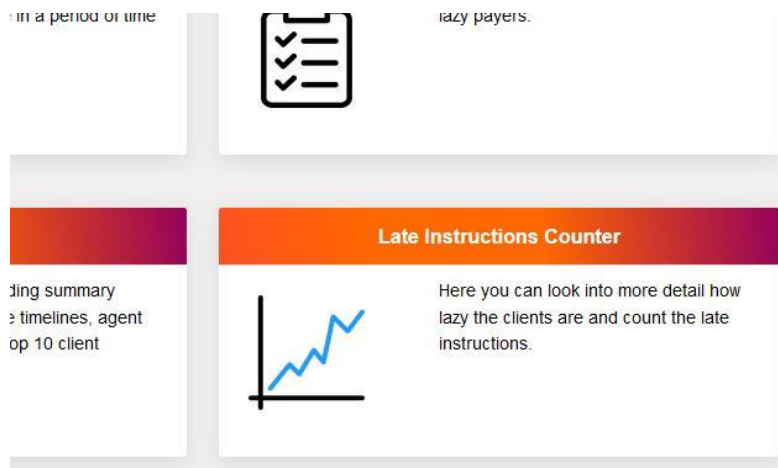


Figure 4.4 – Web2Py portal - Late instr. Counter card

The Python code created in Jupyter Notebook has to be now adapted into Web2PY , since Web2Py works with forms [33] and some of the widgets defined in Jupyter Notebook do not work in Web2Py, because of the implementation of the Python code in the controller and display of information in the view.

The “late_instructions_counter.py” inside the controller contains all the necessary computation. The only thing modified from the Jupyter Notebook code was the part of selecting the client and the filter.

```
form = SQLFORM.factory(
    Field('CLIENTNAME', 'string', widget = SQLFORM.widgets.autocomplete(
        request, db.DIAMS_Companies.Name, distinct = True, limitby=(0, 15), min_length=2)
    ),
    Field('MONTHSYEARS', 'boolean'),
    Field('MONTHSYEARS', requires=IS_IN_SET
        (["-- All --", "1991-2000", "2001-2010", "2011-2020", "2021-2030", "2000", "2001", "2002",
        "2003", "2004", "2005", "2006", "2007", "2008", "2009", "2010", "2011", "2012", "2013", "2014",
        "2015", "2016", "2017", "2018", "2019", "2020", "2021"]), default="2000"
    )
)
```

Figure 4.5 – late_instructions_counter.py - Web2Py form

In Figure 4.5 a code snippet is shown on how the passed form from the controller to the view is done with Web2Py. This form replaces every code that was needed to create the filters for the client and data in the notebook.

The “late_instructions_counter.py” file’s main function named *LateInstructionsCounter* returns the code from Figure 4.6, which are the objects passed to the view. This function does the query computation depending on the selected filters.

```
return dict(
    form=form, value=value, chart_Info=sqlreslist, chart = fig, clientName = form.vars.CLIENTNAME,
    monthsYears = form.vars.MONTHSYEARS, noresultfilter=noresultfilter
)
```

Figure 4.6 – late_instructions_counter.py - Return for view

The “late_instructions_counter.py” also contains another function for the creation of the chart named “createChart(sqlgenerated, filter)”, where the first value passed with the function is the list with the resulting records of the query and the filter is the selection of years or months. This function is called within the main function of the controller and

returns an image. This image is then passed forward with the return in Figure 4.5 to the view.

From the return function in Figure 4.6, the different widgets were individually added inside the view created. This was done by adding Python code in the HTML file by using “{ }” like in Figure 4.7.

```
...
<div class="col-xs-4 col-sm-4 col-md-4 col-lg-4" style="margin:auto;line-height: 2.5em;" >
  {=form.custom.begin}
  <p><b> Client : </b>{=form.custom.widget.CLIENTNAME}</p>
  <p>
    <b> Select a range : </b>
    </p>
    <label class="label">
      <div class="label-text-left"><b>Months</b></div>
      <div class="toggle">
        {=form.custom.widget.SMONTHSYEARS}
        <div class="indicator"></div>
      </div>
      <div class="label-text-right"><b>Years</b></div>
    </label>
    <p>{=form.custom.widget.MONTHSYEARS}</p>
    <p>{=form.custom.submit}</p>
    {=form.custom.end}
  </div>
  ...
```

Figure 4.7 – LateInstructionCounter.html - Passed parameters,controller to view

Figure 4.7 shows how the filters for the client names and date ranges are added into the view in Web2Py, through the return from Figure 4.6. In addition to this, some more features were added directly via JavaScript into the view file. An example of such a feature is when leaving the client’s name blank, the client name input box would be filled with the text “–All–”, to use every available client for the chart generation. Every error that came up could easily be researched on the internet due to the huge Python and JavaScript community available [29].

4.1.2 Deliverable 2 – TM client report

So as to test the tool and start the creation of the report, I could directly access the development server where the tool was running After opening eclipse for the B.I.R.T. tool, two existing reports were copied in order to have a starting design, which was directly after changed to the design shown in Figure 2.3 adapted to our needs. When doing a right click on the report page, and in the insert tab, we could see everything that could be added to the report. In addition to this, objects could also be moved or deleted from the report.

The report design that was used as a reference was not completely different from the needed design, so it was adapted. Inside the main report (Figure 4.8) on the left side, we have the title of the values to display, and on the right side, we have the field that will contain the according result for a specific client.

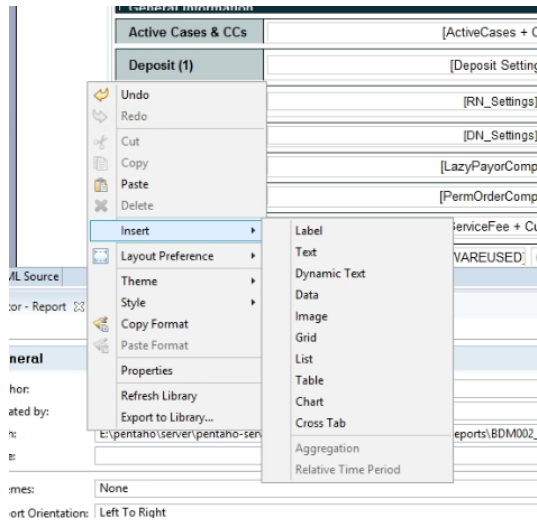


Figure 4.8 – B.I.R.T. tool - Report creation objects

The designs of the reports were changed to look like Figure 4.7 and Figure 4.8.

On the right side of the B.I.R.T. tool after opening a report, we have data sources and data sets (Figure 4.13), among others. We first need to open data sources and establish the connection with the SQL Server of our choice. Next, A data set for every queried field inside the report was created.

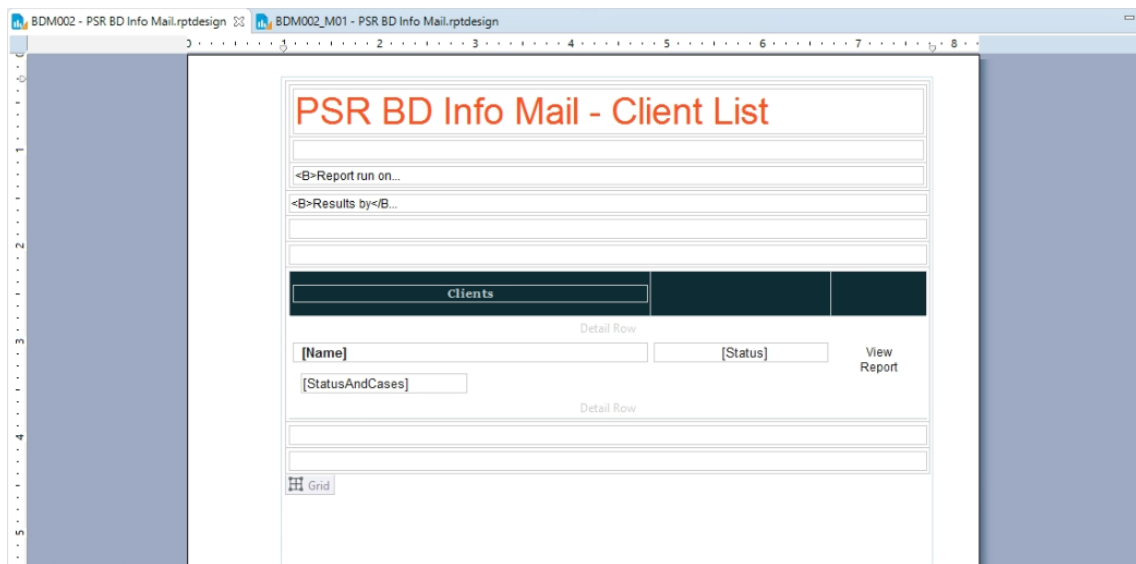


Figure 4.9 – B.I.R.T. tool - Client list report layout

The client list report (Figure 4.9) is displayed after entering a parameter for the client name when running the report, containing a list of clients similar to the previously entered parameter. In order to be able to enter a parameter for this report to be populated with the adequate client list, in the right bar of the client list report, we need to define the “ Report Parameter” named CustomerName. In our case, we want to be able to input a client name when running the report, before showing the Client List report.

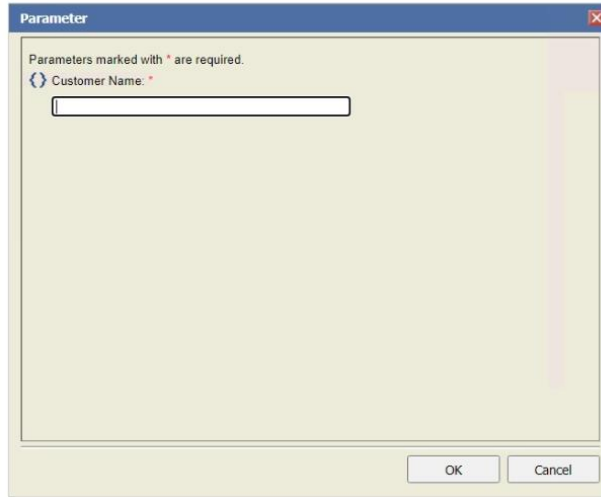


Figure 4.10 – B.I.R.T. tool - Enter client list report parameter

Because of the adding of the new “Report Parameter”, when running the Client list report, we get the window from Figure 4.10 where we need to enter a customer name. Now, we need to transform the entered client name into a “SQL-like parameter” and pass this parameter to the query that populates the report. This is done over the data set we need to create for the population of the report information (Figure 4.10).

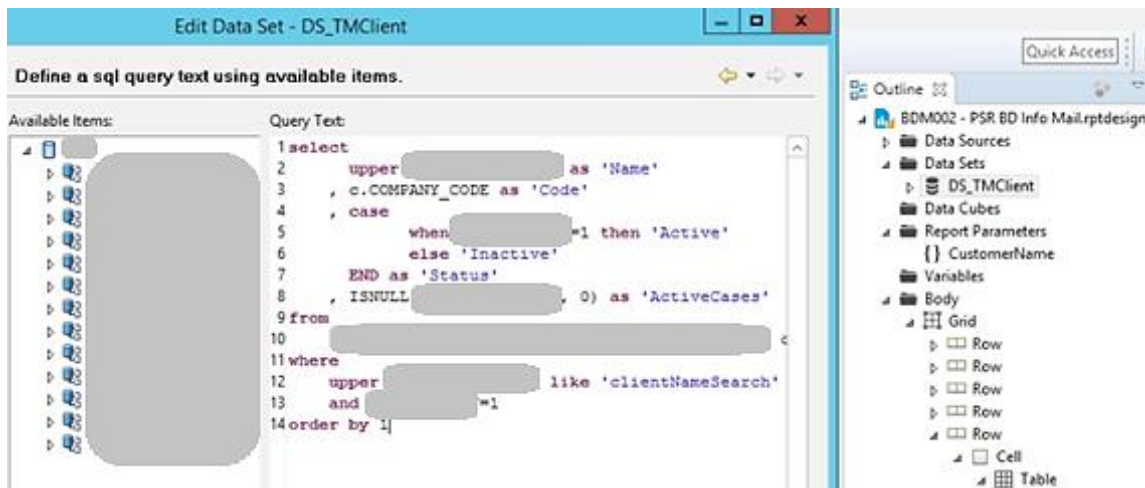


Figure 4.11 – B.I.R.T. tool - Client list report data set creation

The data set in Figure 4.11 contains the query to populate all the clients that are similar to the client name we entered in the Parameter windows for the customer name. In the like part we add ‘clientNameSearch’ and afterwards we need to define a JavaScript transformation for this parameter by adding the following code into the “before open” property of the “DS_TMClient” data set, like it is displayed in Figure 4.12.

```
var paramName = "%" + params["CustomerName"].value + "%";
paramName = paramName.replace("'", "''").replace(" ", "%");
this.queryText = this.queryText.replaceAll("clientNameSearch", paramName);
```

Figure 4.12 – B.I.R.T. tool - Data transformation JavaScript

This JavaScript code transforms the entered name in order to be added to the query in Figure 4.12.

Now we have arrived at the point where we can populate the Client list report. However, in Figure 4.9 we see a field “View report”, which will be available for every returned record in this client list report. By clicking on this report, we want to populate that specific Trademark client report from Figure 4.14. To make the “View Report” label from the client list report clickable, we need to select it and the hyperlink tab under hyperlink options with the specific values we want to forward (Figure 4.13), we need to select the other report created, such that when we click on one specific “View report” label it triggers the run of the trademark client report from Figure 4.9 and forwards client name and client code.

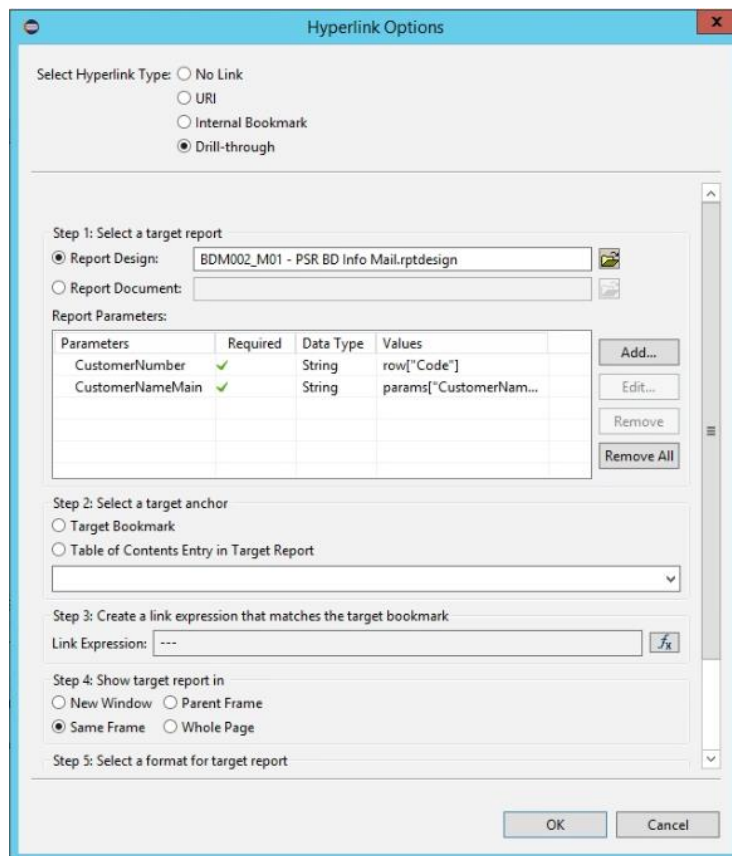


Figure 4.13 – B.I.R.T. tool - Client list report hyperlink options

For the second report which is the report containing the main information, the same principle is applied. We create the datasets (Figure 4.15) with the queries we want to use to query specific user information and add it to the different report labels by doing a right click on the field → Change data column → Select data set & select the value to show (Figure 4.17)



Figure 4.14 – B.I.R.T. tool - Client report layout

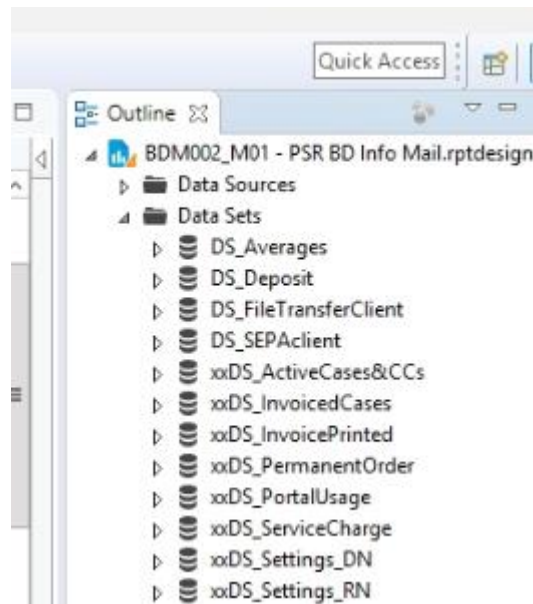


Figure 4.15 – B.I.R.T. tool - Data Sets

One of the many queried fields inside the report is if the clients want their invoices printed. We create a new dataset, name it according to the information we want to query and after opening the data set, we see the window as in Figure 4.16. We add the according query on the right side, to query the field from the database telling us that information and finish this up by clicking on “ok”.

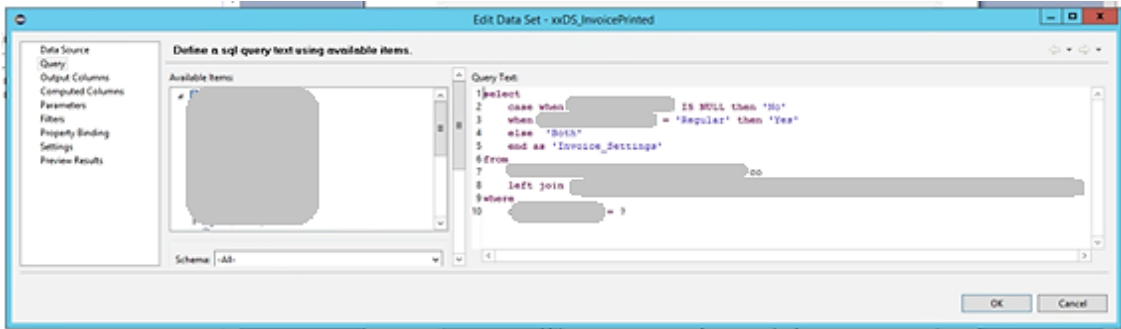


Figure 4.16 – B.I.R.T. tool - Invoice printed data set

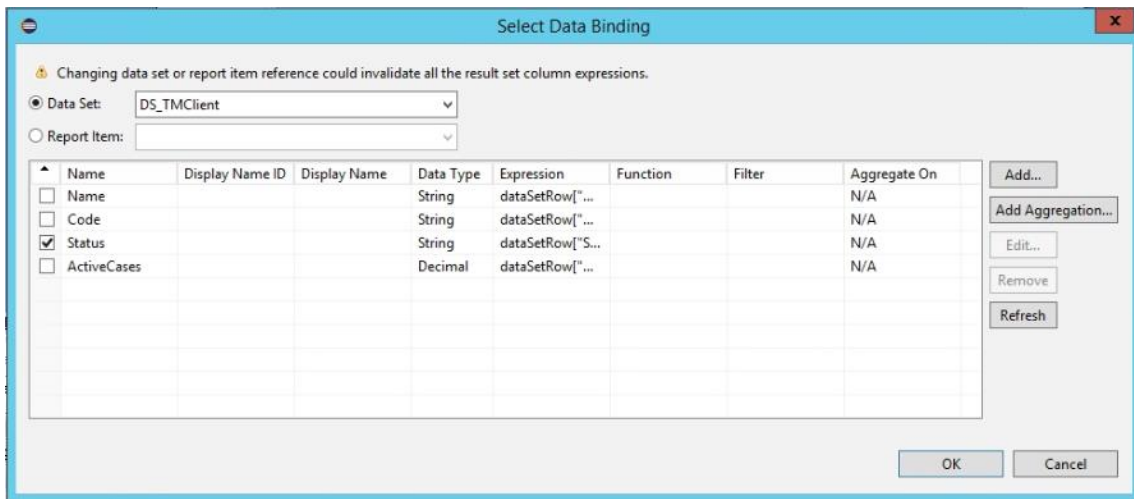


Figure 4.17 – B.I.R.T. tool - Data Set and value selection for field

In Figure 4.17 you can see that if we want to bind a specific data set to a report field, we need to select the previously created dataset with the query in the “Data Set:” field underneath the report which may populate more rows in the “Data column binding”. Afterwards, to select exactly what should be displayed in that specific field, still in the same Figure shown, we click on “Edit” and then on expression, opening up the “Expression Builder” shown in Figure 4.18.

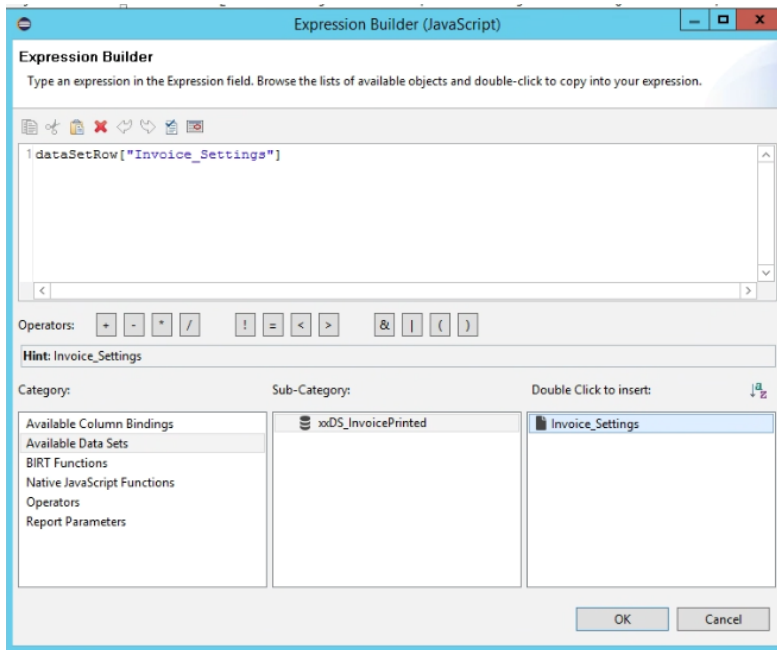


Figure 4.18 – B.I.R.T. tool - Expression Builder

This Expression builder (Figure 4.18) is used to define the result of the specific fields. With this, we are also able to calculate and concatenate different records into one and display them under one label.

4.1.3 Deliverable 3 – Merging of portals

Since it has been decided that the Web2Py portal should only be accessible through the Pentaho portal, the Pentaho portal was studied in order to add a new tab for the CIU portal just like in the draft shown Figure 3.9. An existing and unused tab for “Updates” was simply replaced by a CIU tab like it is shown in Figure 4.19 in the “Mantle.jsp” file of Pentaho.

```

<!-- <li class="nav-li" title="News and Updates">
  <span class="fui-list-columned"></span>
  <div>Updates</div></li> -->

<li class="nav-li" title="Access CIU Portal">
  <span class="fui-list-columned"></span>
  <div>CIU</div></li>

```

Figure 4.19 – Pentaho portal - CIU tab

This tab was already associated with the empty and unused Updates.jsp file. Consequently, to show the Web2Py portal inside the iframe when selecting the CIU tab, the following like was added to the “Updates.jsp” file:

```

<metabhttp-equiv="refresh" content="time";
URL=http://127.0.0.1:8000/portal_local/default/user/login?<%=request.getRemoteUse
r()%>@dennemeyer.com"/>

```

This line calls the URL above mentioned in an iframe associated with the CIU tab and by adding the JSP code “<%=request.getRemoteUser()%>” inside the URL, we can tell which user is currently logged in to Pentaho and pass the username automatically over the URL for Web2Py.

Now, for this to work we also need to change the way the Web2Py Login page is handled by bypassing the login procedure in the Web2Py portal. This is done with the JavaScript code seen in Figure 4.20.

```
var user = (window.location.href).split('?')[1];
$('#auth_user_email').text(user).val(user);
$('#auth_user_password').text('ahd$8d7s#7s3sjxsmxylqps!2=Dwjds!csdfhfgsdqQ3$').val('ahd$8d7s#7s3sjxsmxylqps!2=Dwjds!csdfhfgsdqQ3$');
$('#btn-primary').click(
);
```

Figure 4.20 – Web2Py portal - Bypass login script

The code used to bypass does the following, the login page extracts the username from the previously passed URL and puts it in the username field of the login. In addition to this, the script also inputs the master password a colleague implemented for every user in the Web2Py user’s table, which is the same for everyone. Finally, the code simulates a user clicking on the submit button on the login page. This is how the bypassing of the login was done and the user only needs to log in once in the main portal which is Pentaho.

The last issue to solve is to make the Web2Py inaccessible from outside the Pentaho portal. This was also done from inside the user.html page by adding to it the script in Figure 4.21. This script redirects a user trying to access the Web2Py portal directly to the Pentaho main page by checking from where the Web2Py is being opened. This is done by comparing the “window.location” parameter with the “window.parent.location” parameter, since the Web2Py portal can only be accessed through an iframe in the Pentaho portal.

```
if ( window.location !== window.parent.location )
{
    alert("Correct call of ciu portal")
}
else
{
    alert("Incorrect call of ciu portal")
    window.location.replace("https://127.0.0.1/BusinessIntelligence/Home");
}
```

Figure 4.21 – Merged portals - Block Web2Py access from outside Pentaho

4.1.4 Deliverable 4 – Study on visualization tools

The final objective, the study of the data visualization tools was tackled by installing the trial version of each tool, and by knowing the existing functionalities of the portals, trying out if the visualization tools can replace and reproduce these functionalities the portals have. A set of functionalities were defined to study the capabilities of these tools related to the department's needs.

4.2 Testing process

4.2.1 Deliverable 1 – Late instructions chart

The first deliverable is the implementation of a report on the CIU portal and was first created and tested in Jupyter notebook and approved by the supervisor of the internship and the corresponding departmental manager. The queries were created and tested in SSMS and afterwards sent as well to the supervisor of the internship for correction and approval. After having the queries approved, the chart was created. Afterwards, it was accordingly modified to work inside my local web2py version of the portal.

Finally, integration tests, acceptance tests and behavioral tests have been performed for every software deliverable.

Integration testing has been performed after adding the analytical module to the existing local Web2Py portal. It was expected that no conflicts appear during the search, opening and execution of the module. Over the course of the project some dozen tests were done to see if the integrated analytical module doesn't cause any conflicts with other existing modules. The obtained results were the expected ones, such as no existing conflicts with other modules and the correct display of the module.

Behavioral testing was also performed, expecting the module to return correct and reliable information in the displayed chart when executing the module. When running the module, the chart with the correct late instruction information for a specific client in addition to a table containing the amounts for these late instructions are displayed.

Lastly, acceptance tests were done, expecting the module to be reactive, the UI to be clean and stay clean on execution and finally for it to be easy to use. The UI stays clean on execution and is reactive, not taking long to load. Additionally, module fulfils its purpose of being easy to use, by not having many choices in the filter part of the module.

All the tests were done nearly a dozen times by my project supervisor and by a data engineer colleague of the department with the knowledge in Web2Py and by me, to make sure of the correct development and integration of the analytical module.

4.2.2 Deliverable 2 – TM Client report

The testing done for the trademark client report was done from inside the development server. Having SSMS open in parallel to the report builder, allowed me to quickly control the queries I created and added into the report by comparing the results from SSMS with the results from the report. Moreover, the queries were also sent to another colleague from the CIU department with a certain expertise in the trademark domain, additionally controlled and tested if the queries return the required information.

Integration testing has been performed after adding the analytical report to the existing Pentaho portal. It was expected that no conflicts appear during the search, opening and execution of the report. Over the course of the project dozens of tests were done to see if the integrated analytical report doesn't cause any conflicts with other existing modules. The obtained results were the expected ones, such as no existing conflicts with other modules and the correct display of the module.

Behavioral testing was also performed, expecting the module to return correct and reliable information in an easily human readable format. When running the module, it returns a readable report for a specific client

Lastly, acceptance tests were done, expecting the module to be reactive, the UI to be clean and stay clean on execution and finally for it to be easy to use. The UI stays clean on execution and is reactive, not taking long to load the report. The report is also very clean and is correctly displayed no matter what information is generated for a specific client. Additionally, the module fulfils its purpose of being easy to use, by being straightforward and not having many options to choose from.

All the tests were done nearly a dozen times by my project supervisor and by a data engineer with knowledge in B.I.R.T and Pentaho and by me, to make sure of the correct development and integration of the analytical module.

4.2.3 Deliverable 3 – Merging of portals

The first implementation of the merging of the portals was tested from inside the development server. Generic tests were done to see if we could access the Web2Py server directly and to control if the redirection works. Additionally, the login was controlled, by logging in once and trying to generate different Pentaho and Web2Py reports, while staying on one interface.

Finally, several additional quality tests were done from inside Pentaho and Web2Py to ensure that we were obtaining the expected behaviors, that backwards compatibility was still functioning, and that the performance had not dropped in a significant manner.

Integration testing has been performed after implementing the merging of the portals. It was expected that this implementation doesn't bring any conflicts during the search, opening and execution of any analytical module from any portal. A dozen tests were done and the obtained results were the expected ones, there are no conflicts when accessing or executing any existing analytical module from any portal.

Behavioral testing was also performed, expecting the implementation to behave as follows, when trying to access the Web2Py portal directly, we should get redirected to the Pentaho portal and login, in order to access analytical modules of the Pentaho portal and the modules of the integrated Web2Py portal. It is also expected that when accessing the Web2Py portal from inside Pentaho, that a log in to Web2Py is automatized for the user to see the Web2Py reports. A dozen tests were done, trying different modules, and accessing the Web2Py portal directly. The portals behave the way they should and every module gets displayed correctly without any considerable delay.

Lastly, acceptance tests were done, expecting the merged portals to be reactive, the UI to be clean and stay clean on execution and finally for the merging solution to be easy to use. The UI for the integrated Web2Py portal inside Pentaho stays clean when called upon and is reactive and logs in without any delay, showing the content of the Web2Py portal directly. The interface for the integration of the Web2Py portal is also clean and is correctly displayed, giving the user the impression that he is still on the same Pentaho portal. Additionally, the implementation is quite straightforward, when clicking on the CIU tab of the Pentaho portal, the iframe on the right is populated with the integrated Web2Py portal and the login is done automatically.

All the tests were done nearly a dozen times by me, in addition to my project supervisor, data engineer and data engineer with knowledge in both portals, to make sure of the correct implementation of the merging.

Chapter 5:

Results

5.1 Validation process

5.1.1 Deliverable 1 – Late instruction chart

In Figure 5.1 the final and version of the late instruction counter is displayed, which was put live in the production server and available for internal usage by the trademark department.

After finishing the first module, the new code was pushed via Git from the local machine into the development server for validation by the supervisor of the internship and the corresponding management. Their knowledge in the trademark domain and in the late fees for trademarks allowed them to test the tool and analyze the results given quickly and efficiently by also locating errors or seeing weird inconsistencies in the results. In addition, the supervisor of the internship also tested and analyzed the queries used manually, over SSMS, which allows him to have a better overview of the data selected and passed.

Late Instructions Counter



Figure 5.1 – Live version Late Instr. Counter (web2py portal)

After selecting the mandatory fields and running the module, the chart and the table are shown a few seconds later.

It was initially defined that the tool should only mainly accomplish its purpose, that is showing the chart and the table with the results, therefore not too much attention was paid to the design of the module, but of course, the chart is clean, and the design has also been validated by the supervisor of the internship and the whole CIU team. In addition, to help with the validation, the module was also presented in a meeting with all the members of the CIU to get different opinions on how it should look, consequently, the feedback was received that the design is clean and simple, how it is supposed to be

To conclude the validation, randomly chosen clients and date ranges were selected and analyzed, before giving the go to push this new module into the staging server, and finally to the production server containing the live version of the portal.

Validation of the data is also counter checking several data set and performing plausibility analysis.

Furthermore, randomly chosen clients and date ranges were selected and analyzed, before giving the go to import the module in the Pentaho portal of the production server.

To conclude the validation, the green light was gotten from the supervisor of the internship to push this newly developed module into the staging server, and finally to the production server containing the live version of the portal. The stakeholder responsible for the final approval of the module was the company supervisor.

This tool is now available and per default, the trademark department has access to this tool for their analytical purposes, being an asset that will allow getting the late instruction information about every Trademark client in a few seconds, with the objective to determine how the clients are behaving concerning the company's trademark renewal services.

5.1.2 Deliverable 2 – TM Client report

For the validation of the second deliverable, which is the Trademark client report, a few points were defined by the supervisor of the internship. The report should be clean and easily readable, contain certain client information and finally, the information shown should be correct.

Their knowledge in the trademark domain and in the late fees for trademarks allowed them to test the tool and analyze the results given quickly and efficiently by also locating errors or seeing weird inconsistencies in the results. The TM client report was mainly validated by the supervisor of the internship, by testing and analyzing the queries used manually, over SSMS, which allows him to have a better overview of the data selected and passed.

A design was initially drafted (Figure 3.8) and validated by the supervisor of the internship, as was the final design from Figure 5.3.

In Figure 5.3 the final version of the TM client report is displayed, which was put live in the production server and available for internal usage by the trademark department.

BIRT Report Viewer
 Showing page: 1 of 1
 Denнемeyer
 The IP Group
PSR BD Info Mail - Client List
 Report run on: December 08, 2021 at 03:13 PM
 Results by: test

testclient1 Client Code: test1 633 cases	Active	View Record
testclient2 Client Code: test2 0 cases	Inactive	View Record
testclient3 Client Code: test3 1 cases	Active	View Record
testclient4 Client Code: test4 0 cases	Inactive	View Record
testclient5 Client Code: test5 1 cases	Active	View Record
testclient6 Client Code: test6 10094 cases	Active	View Record

Figure 5.2 – Live version TM Client report - Client selection (Pentaho portal)

When running the report, we get a window to select or write the name of the desired client we want to analyze. After submitting the name, for example, “test client”, we get the intermediary report as shown in Figure 5.2 with all the similar client names. After selecting a client from this report, the date is entered in a window that pops up, and finally, the TM client report is shown as displayed in Figure 5.3.

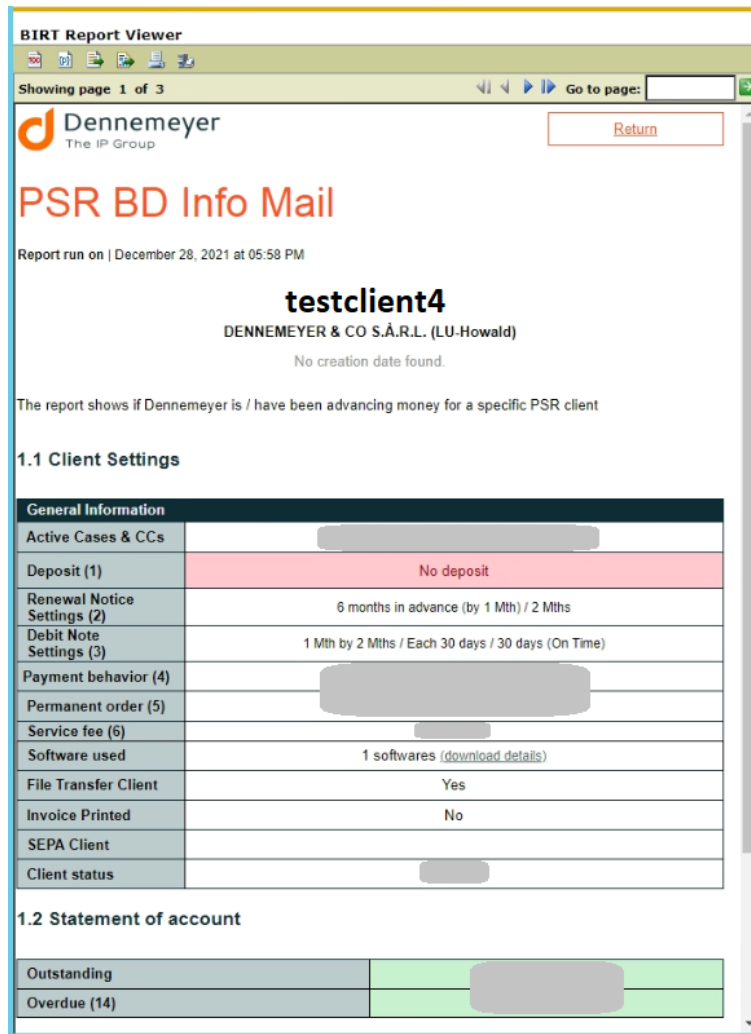


Figure 5.3 – Live version TM Client report - Report (Pentaho portal)

Furthermore, randomly chosen clients and date ranges were selected and analyzed, before giving the go to import the module in the Pentaho portal of the production server.

To conclude the validation, the green light was gotten to import the report into the Pentaho portal in the development server. No version control was used for this report because currently, the Pentaho portal is not being tracked. The stakeholder responsible for the validation of this Pentaho report was the project supervisor.

Finally, the report is now available and per default, the trademark department has access to this tool for their analytical purposes, being an asset that will allow getting the most important information about any trademark client in a few seconds, with the objective to have at hand a clear and simple client identity report.

5.1.3 Deliverable 3 – Merging of portals

For the validation of the third deliverable, the merged solution was presented to the CIU team and approved. Additionally, the supervisor of the internship tested the solution for

bugs and the user experience is fine. The integration of the Web2Py portal into the Pentaho portal looks good. Additionally, the single login for the usage of the portals works without any issues after some colleagues from the CIU team tested it.

In Figure 5.4, the final version of the merged portals is shown, where we can clearly see the Web2Py portal open inside the Pentaho portal. They are still two independent portals, but the impression we get is that it's a single portal.



Figure 5.4 – Merged portals

When clicking on the small CIU tab on the left side of the Pentaho portal, we get automatically logged in to the Web2Py portal, so this transition is not visible to the user and does not interrupt the search for a desired analytical module inside any portal.

Furthermore, random logins and logouts of the user have been done, and different Web2Py charts and reports have been generated and tested from inside the Pentaho portal.

To conclude the validation, the green light was gotten to git push the changes done in the code of the Web2Py portal, from the development server into the staging and then to the production server. Additionally, the changes made to the code of the Pentaho portal were copied from the Pentaho portal in the development server, into the staging server, and then into the production server, since there is no version control in place for the Pentaho portal. The stakeholder responsible for the final approval of the implementation of the merged portals is the project supervisor.

Finally, the merging has been concluded and every user can access both portals just by going through the Pentaho portal to log in once. This implementation is an asset to every person using the portals since this removes the hassle of logging in into two portals, every module is under one interface and now the users do not need to be managed manually.

Chapter 6:

Conclusion

To summarize, two data analytical modules were created inside the two working portals for the CIU department of Dennemeyer. The first of them was created with Python and integrated and adapted to Python code working with Web2Py, where the Python code needed to be adapted to a controller and a view. This Web2Py module contributes to the analytical part available for trademark clients, showing how much they've been late instructing over the years and months.

The second module created has been done inside the B.I.R.T. tool in eclipse, which was afterwards imported into the second portal which is Pentaho. This second module brings an overview report of a specific client, showing their most important information, just a few clicks away.

The third and most critical part of the project was the merging of the portals which was solved by embedding the current Web2Py portal into an iframe displayed from inside the Pentaho portal. The merging of the portals discards the load the CIU department has in creating all the new users and additionally, users only need to login once for the usage of both portals that are under one interface and not two interfaces, as it was before.

The code developed in the three first deliverables was put live for internal usage after being tested and validated.

Finally, a study on three different data visualization tools was done, however focusing most of the studying time on Power BI since the company already has licenses for the Power BI pro version. This study gives insight on the different tools, on how powerful they are, and from a feature list defined, see which tools support the defined capabilities. This will help the CIU team choose a new tool for their data visualizations.

The two modules developed for both portals are currently running in the production environment, in addition to the implementation of the merged portals.

6.1 Difficulties encountered

The first challenge was to gather experience in python programming since Web2Py mainly works with Python, however, the next challenge came directly after when implementing the module in Web2Py since many of the widgets couldn't be simply copied into the Web2Py, since Web2Py works with Models, views and controllers.

Regarding the implementation of the Pentaho report, starting from zero with a new challenge, but I had a few calls with a former colleague where he explained to me the basics of the B.I.R.T. tool. Knowing SQL and JavaScript also allowed me to understand the tool and get the hang of it more easily.

In the creation of the two first modules, there was one constant similar challenge, which was to query the right information for the modules. Many calls and discussions had to be held in order to ensure that the information queried is correct and good for usage.

Passing on to the merging of the portals, the main challenge was finding a way to merge both portals, until a simple embedding of Web2Py in Pentaho was decided. Creating a new CIU tab for the Web2Py integration in the Pentaho portal was also being complicated to do until it was simply decided to replace an unused tab for the Web2Py portal integration. In addition to this, logging in once to access both portals was also a complicated task to do because we were thinking way too complicated by trying to make everything run in one service without having both Pentaho and Web2Py running. The Pentaho community is very small to practically non-existing, consequently finding any other way to merge the portals was nearly impossible without having years of expertise with both portals.

6.2 Future work

Related to the modules that can be created for both portals, there are lots of possibilities. Many different modules can be created containing different information, adapted for every department in the company.

There are also quite some possibilities for the future of the merged portals.

One is the complete integration of the web2py portal into the Pentaho portal or the other way around could be achieved in the future. However, this would take lots of work, time and patience, since all the existing dashboards and reports would have to be recreated and adapted to the other portal and their used technology. This works both ways, adapting Pentaho reports to the Web2Py portal and Web2Py or adapting the Web2Py reports to Pentaho and its' used technology.

Alternatively, the complete integration of the reports into a new data visualization tool like the ones studied in the last deliverable would be a good option. For example, after having studied 3 data visualization tools in deliverable 4, it was concluded that most of the reports available in Pentaho and web2py could be recreated in Power BI. In addition, Power BI supports most of the feature requirements that we had defined for the data visualization tool study.

For the merging of the portals part, changes could be made the way the automatic login into the CIU portal is done when logging in from the Pentaho portal. Finding a way to encrypt the master password used for our solution, so that it cannot be found and viewed in the inspector of internet browsers like Mozilla Firefox. Naturally, additional analytical modules, reports, and dashboards could be implemented into both portals to further expand the list of analytical features available.

Finally, there is the possibility of updating the structure of the current portals with a more modern .NET framework, while delegating the visualization part to Power BI.

References

- [1] Anaconda | Individual Edition. (2021). Retrieved 13 December 2021, from <https://www.anaconda.com/products/individual>
- [2] BIRT. (2021). Retrieved 29 December 2021, from <https://eclipse.github.io/birt-website/>.
- [3] Creating a Power BI Dashboard - A Step by Step Guide | InsightWhale. Digital Analytics, Conversion Rate Optimization and Business Intelligence. (2021). Retrieved 29 December 2021, from <https://insightwhale.com/creating-a-power-bi-dashboard-a-step-by-step-guide/>.
- [4] CSS Tutorial. (2021). Retrieved 13 December 2021, from <https://www.w3schools.com/css>
- [5] Das, S. (2021). Why Jupyter Notebooks Are So Popular Among Data Scientists. Retrieved 13 December 2021, from <https://analyticsindiamag.com/why-jupyter-notebooks-are-so-popular-among-data-scientists/>
- [6] Data Visualization | Microsoft Power BI. (2021). Retrieved 28 December 2021, from <https://powerbi.microsoft.com/en-us/>.
- [7] Flowchart Maker & Online Diagram Software. (2021). Retrieved 13 December 2021, from <https://app.diagrams.net>
- [8] Git. Git-scm.com. (2021). Retrieved 29 December 2021, from <https://git-scm.com/>.
- [9] How to Connect Python to SQL Server using pyodbc - Data to Fish. (2021). Retrieved 13 December 2021, from <https://datatofish.com/how-to-connect-python-to-sql-server-using-pyodbc>
- [10] Importance, Purpose, and Benefit of Data Visualization Tools!. (2021). Retrieved 17 December 2021, from <https://splashbi.com/importance-purpose-benefit-of-data-visualization-tools/>
- [11] Installation — Matplotlib 3.4.3 documentation. (2021). Retrieved 13 December 2021, from <https://matplotlib.org/3.4.3/users/installing.html>
- [12] Introduction to HTML. (2021). Retrieved 13 December 2021, from https://www.w3schools.com/html/html_intro.asp
- [13] Ipython Notebook Tutorial. (2021). Retrieved 13 December 2021, from <https://plotly.com/python/ipython-notebook-tutorial/>
- [14] JavaScript Tutorial. (2021). Retrieved 13 December 2021, from <https://www.w3schools.com/js>

- [15] Matplotlib - Visualization with Python. (2021). Retrieved 13 December 2021, from <https://matplotlib.org>
- [16] Pentaho from Hitachi Vantara. SourceForge. (2021). Retrieved 29 December 2021, from <https://sourceforge.net/projects/pentaho/>.
- [17] Project Jupyter. (2021). Retrieved 13 December 2021, from <https://jupyter.org>
- [18] Pyplot tutorial - Matplotlib 3.5.0 documentation. (2021). Retrieved 13 December 2021, from <https://matplotlib.org/stable/tutorials/introductory/pyplot.html#sphx-glr-tutorials-introductory-pyplot-py>
- [19] Performance Testing Tutorial: What is, Types, Metrics & Example. (2021). Retrieved 13 December 2021, from <https://www.guru99.com/performance-testing.html>
- [20] Qlik | Data Analytics & Data Integration Solutions. (2021). Retrieved 28 December 2021, from <https://www.qlik.com/us/>.
- [21] Qlik Sense Tutorial For Beginners – Features and Architecture. (2021). Retrieved 28 December 2021, from <https://data-flair.training/blogs/qlik-sense-tutorial>.
- [22] Qu'est-ce qu'Azure Boards ? Outils de gestion des projets de développement de logiciels. - Azure Boards. (2021). Retrieved 13 December 2021, from <https://docs.microsoft.com/fr-fr/azure/devops/boards/get-started/what-is-azure-boards?view=azure-devops>
- [23] Self-Service Analytics, Data Science & Automation | Alteryx. (2021). Retrieved 28 December 2021, from <https://www.alteryx.com/>.
- [24] Smith, D. (2021). Alteryx integrates with Microsoft R. Revolutions. Retrieved 29 December 2021, from <https://blog.revolutionanalytics.com/2017/03/alteryx-integrates-with-microsoft-r.html>.
- [25] Sprints | Atlassian. (2021). Retrieved 13 December 2021, from <https://www.atlassian.com/agile/scrum/sprints>
- [26] SQL Server Downloads | Microsoft. (2021). Retrieved 29 December 2021, from <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>.
- [27] SQL Server Management Studio (SSMS) - SQL Server Management Studio (SSMS). (2021). Retrieved 13 December 2021, from <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15>

- [28] SSMS example. (2021). [Image]. Retrieved from (2021). [Image]. Retrieved from https://www.softpedia.com/get/Internet/Servers/WEB-Servers/Microsoft-SQL-Server-Management-Studio-Express.shtml#sgal_0
- [29] Stack Overflow - Where Developers Learn, Share, & Build Careers. (2021). Retrieved 28 December 2021, from <https://stackoverflow.com/>.
- [30] Swim Lane Diagram: A Complete Guide to Using Swim Lane Diagrams | MindManager. (2021). Retrieved 13 December 2021, from <https://www.mindmanager.com/en/features/swim-lane-diagram>
- [31] The Basics of a Microsoft SQL Server Architecture - INAP. (2021). Retrieved 13 December 2021, from <https://www.inap.com/blog/microsoft-sql-server-architecture/>
- [32] The Starter Guide to Data Visualizations. (2021). Retrieved 21 December 2021, from <https://www.klipfolio.com/blog/starter-guide-to-data-visualizations>
- [33] web2py - Forms and validators. (2021). Retrieved 13 December 2021, from <http://web2py.com/books/default/chapter/29/07/forms-and-validators>
- [34] Web2py – Web Framework (2021). Retrieved 28 December 2021, from <http://web2py.com/>.
- [35] What is the Jupyter Notebook? — Jupyter/IPython Notebook Quick Start Guide 0.1 documentation. (2021). Retrieved 13 December 2021, from http://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html
- [36] What is a Fishbone Diagram? Ishikawa Cause & Effect Diagram | ASQ. (2021). Retrieved 29 December 2021, from <https://asq.org/quality-resources/fishbone>.
- [37] What Are Azure Boards And Why Is It Important? - Whizlabs Blog. (2021). Retrieved 13 December 2021, from <https://www.whizlabs.com/blog/what-are-azure-boards-and-why-is-it-important/>
- [38] What Is Agile Methodology in Project Management?. (2021). Retrieved 13 December 2021, from <https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/>
- [39] What Is CI/CD and How Does It Work? | Synopsys. (2021). Retrieved 13 December 2021, from <https://www.synopsys.com/glossary/what-is-cicd.html>
- [40] What Is Data Visualization? Definition, Examples, And Learning Resources. (2021). Retrieved 21 December 2021, from <https://www.tableau.com/learn/articles/data-visualization>

- [41] What is Microsoft SQL Server? A definition from WhatIs.com. (2021). Retrieved 13 December 2021, from <https://searchdatamanagement.techtarget.com/definition/SQL-Server>
- [42] What is Power BI? - Power BI. (2021). Retrieved 28 December 2021, from <https://docs.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview>.
- [43] Why You Need Agile Feedback Loops During & After Sprints. (2021). Retrieved 13 December 2021, from <https://www.mendix.com/blog/agile-process-why-you-need-feedback-loops-both-during-and-after-sprints/>