Human Resource Management Aid Tool

Pedro Nuno Courela Martins

Mestrado em Engenharia Informática

Trabalho de Projeto orientado por:
Prof. Doutor Nuno Ricardo da Cruz Garcia
Acknowledgments

The first words go to my parents and brother. This work is the culmination of a long journey which, without them would be impossible. Thank you for all the love, support, effort and example of life. All my conquests and victories are the fruit of their support.

I would like to thank my coordinator, Prof. Nuno Garcia, for the guidance he provided, his availability, effort and patience.

This work was done at InnoWave, a company where I was already working before doing this thesis, I would like to thank the CEO Tiago Gonçalves and all the people involved in letting me do my thesis at the company, for putting their trust in me and being always available to provide support when needed.

I would also like to give special thanks to Tiago Correia and Daniel Pinto, who were there since the beginning, giving me orientations and tips about the steps to take throughout the work.

This achievement would not have been possible with all the support of my family and friends who helped me in this journey. To my friends and colleagues who accompanied and helped me throughout the journey at the Faculdade de Ciências da Universidade de Lisboa.

Finally, thank you to all my family and all the people who, in some way, helped me to achieve this degree.
To my parents and brother
Abstract

My thesis was done at InnoWave which is a consulting company based in Lisbon but with businesses all over the world, due to this fact it becomes complex the task of making the human resources management in the various projects, which is currently done through various tools in the company that do not exchange information automatically. This is the central problem addressed in this thesis, decentralization of information, and which I intend to solve by developing a system that helps managers in their daily work.

Due to the fact that for managers to manage all their employees it is necessary for them to access 3 different platforms and cross-reference the data from the various platforms with manual tasks, often with the help of Microsoft’s Excel tool. This is not at all the most efficient way of working and it was due to this fact that I decided to propose this subject and help the company in its tasks.

The objective of this work is to build a system that can centralize all the necessary information in a single place in an automatic way, thus optimizing the tasks that are currently done by managers and bringing added value with the information acquired.

The solution that I planned, designed and implemented was mostly built using 3 softwares from Power Platform: Power BI, Power Apps and Power Automate. In Power BI it was possible to create dashboards that group the information in a useful way for managers. In Power Apps we created an application that complements the dashboards where managers can edit information about their employees. In Power Automate were created automatic flows of sending emails not only for employees but also for the managers themselves and for the Office department of the company.

Keywords: Human resources management, Decentralization of information, Manual tasks, Tasks automation and optimization, Power Platform
Resumo

A minha tese foi feita na InnoWave que é uma empresa de consultoria sediada em Lisboa mas com negócios por todo o mundo, devido a este facto torna-se complexa a tarefa de fazer a gestão dos recursos humanos nos vários projetos, o qual é feito atualmente através de várias ferramentas existentes na empresa que não trocam informação entre si de forma automática. Este é o problema central abordado nesta tese, descentralização de informação e o qual pretendo resolver desenvolvendo um sistema que auxilie os gestores no seu trabalho diário.

Atualmente para um gestor saber informações sobre os seus empregados tais como em que projeto se encontra atualmente ou qual a sua alocação num determinado espaço de tempo, este tem de aceder a 3 plataformas diferentes: Orange HRM, Salesforce CRM e WeGrow.

Na plataforma Orange HRM é possível um gestor obter dados não só sobre um empregado tais como as capacidades que tem, linguagens que conhece e as timesheets preenchidas como também é possível encontrar os projetos atualmente válidos.

No Salesforce CRM é onde os gestores criam uma oportunidade, que internamente é chamada de projeto ou charge code, tendo os dois o mesmo significado. Este charge code é utilizado para associar as horas de trabalho de um empregado na sua timesheet assim com também para associar a despesas.

O WeGrow é uma nova plataforma interna de gestão de recursos humanos introduzida na empresa no decorrer dos trabalhos, devido a este facto, a tabela criada na minha base apenas contém os valores que irei precisar e não representa a estrutura de dados real, de outro modo a entrega da tese atempadamente era posta em causa.

Após aceder às 3 plataformas o gestor tem de guardar as informações que necessita em algum lugar, a grande maioria dos gestores tende a usar a ferramenta Excel da Microsoft, onde a podem consultar e editar quando necessário.

Para além desta informação, existem empregados que trabalham no estrangeiro, o que significa que trabalhará nas instalações do cliente, onde necessitará de um alojamento e bilhetes de avião. Esta informação apenas existe em e-mails enviados pelo gestor para o departamento Office, a menos que o gestor também a guarde noutro local.

Outra informação importante são as datas finais, estas podem ser de um charge code em que o gestor, quando a data de fim se aproxima, tem de criar um novo, outra é a data
referente ao fim do contrato de um empregado com um cliente que precisa de ser renovado ou então ser alocado a um novo projecto.

Para os empregados no estrangeiro a situação é idêntica, os gestores têm de saber quando termina o contrato de alojamento, se este será renovado ou se o empregado regressará, nesse caso, precisará de um bilhete de avião. Uma forma que os gestores fazem para contornar o problema é criar alertas manuais nos seus calendários e ao fazê-lo, sabem quando a data final está próxima. Todo este trabalho é feito de forma manual e não autónoma, o que é propício a erros humanos e consome bastante tempo aos gestores fazerem estes passos todos.

Com estes problemas em mente, os objetivos do meu trabalho passam por construir um sistema de raiz desenvolvido especificamente para a InnoWave focado em resolver o problema da descentralização de informação e propor uma possível solução para o preenchimento das *timesheets* no devido tempo, tendo em conta o meu sistema e o processo existente.

Todo o trabalho foi dividido em 3 módulos que tentam solucionar os problemas descritos sendo eles: Gestão de recursos humanos de uma SBU, Visão geral de projetos e Processos automáticos, cada um resolve um problema com um escopo diferente.

No *Salesforce*, um projeto que é criado não dá para associar a uma pessoa, alguns gestores tentam contornar este problema indicando o nome da pessoa no próprio *charge codes*, mas este processo não pode ser aplicado em todos os casos, uma vez que existem *charge codes* que são utilizados por mais do que um funcionário. Este problema é resolvido no primeiro módulo onde guardo a ligação entre um empregado e um *charge code*.

Os gestores necessitam de saber quais são os seus projectos actuais e futuros e quais as pessoas que estão a trabalhar neles de uma forma rápida e consisa. É no módulo Visão geral de projectos onde tento resolver este problema. Além disso, o *Salesforce* não notifica o gestor sobre a proximidade da data de fim dos seus projectos, o que mais uma vez exige que os gestores criem manualmente lembretes.

O preenchimento das *timesheets*, apesar de ser uma tarefa relativamente simples, muitos empregados não preenchem a tempo, o que acaba por causar problemas para a empresa, porque as *timesheets* são um processo muito importante para pagar os salários dos empregados atempadamente. Devido à existência deste problema, no último módulo *Processos automáticos*, este é tido em conta e uma solução é proposta.

O sistema que planeei, desenhei e implementei foi desenvolvido, na sua grande maioria, utilizando a *Power Platform*, uma ferramenta da *Microsoft* composta por 4 softwares diferentes *Power BI*, *Power Apps*, *Power Automate* e *Power Agents*. O *Power Agents* não foi utilizado nesta tese devido ao facto de ser uma ferramenta onde é possível criar chat bots com relativa facilidade, componente que não foi idealizada para o meu sistema.

Através do *Power BI* criei várias *dashboards* que irão agregar a informação de forma
automática em matrizes, gráficos e em valores chave para os gestores, deste modo cada gestor consegue obter de uma forma fácil e rápida informações valiosas sobre os seus empregados.

Foi criada uma aplicação utilizando o **Power Apps** que irá complementar as dashboards, nesta aplicação será possível aos gestores adicionarem, editarem ou removerem uma associação entre um charge code e um empregado e também é possível fazer requerimentos de alojamentos ou bilhetes de avião ao departamento *Office* para empregados que irão trabalhar na sede de um cliente no estrangeiro.

Recorrendo ao **Power Automate** foi possível criar fluxos que são despoletados automaticamente quer seja apartir da aplicação ou dependendo da informação existente na base de dados.

De modo a que o meu sistema seja o mais autônomo possível desenvolvi um pequeno programa na linguagem **JAVA**, o qual irá manter a minha nova tabela de associações de projectos atualizada de acordo com as horas introduzidas por cada empregado na sua *timesheet*.

Nesta tese também apresento uma possível solução para o problema existente no atraso da entrega das *timesheets*, que foi baseada num processo existente e na minha nova tabela de associações de projectos.

São descritos todos os os fluxos possíveis no meu sistema assim como todos os testes, deste modo provando que o sistema funciona e faz o que é esperado, todos estes testes foram feitos por mim de forma manual.

No inicio do trabalho foram identificados todos os requisitos funcionais e não funcionais do sistema que são uma parte essencial no desenvolvimento de qualquer sistema, por isso nesta tese, depois de apresentado todo o trabalho é feita uma validação de cada requisito de modo a indicar se este foi cumprido ou não. Existem no total 12 requisitos funcionais e 9 requisitos funcionais, destes 21 requisitos apenas um não é totalmente cumprindo, estes irão ser todos detalhados no conteúdo desta tese.

O trabalho futuro que poderá ser feito neste sistema tem a haver com acrescentar ainda mais informação às dashboards como por exemplo as férias, que no escopo desta tese não foram abrangidas, a integração do meu sistema com as várias plataformas mencionadas, deste modo os gestores poderão experimentar o sistema no seu dia a dia e dar feedback sobre o que pode ser melhorado ou acrescentado, novas dashboards ou ecrãs podem ser adicionados ao sistema com relativa facilidade o que permite que este seja um sistema que esteja em constante crescimento.

---

**Palavras-chave:** Gestão de recursos humanos, Descentralização de informação, Tarefas manuais, Automatização e optimização de tarefas, Power Platform
# Contents

<table>
<thead>
<tr>
<th>List of Figures</th>
<th>xviii</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>xxi</td>
</tr>
</tbody>
</table>

## 1 Introduction

1.1 Context .............................................. 1
1.2 Motivation ........................................... 3
1.3 Goals .................................................. 4
1.4 Structure of the document ......................... 4

## 2 Related Work

2.1 InnoWave’s management tools .......................... 7
   2.1.1 Salesforce CRM ................................... 7
   2.1.2 Orange HRM ...................................... 8
   2.1.3 Finance Department Tool ......................... 9
2.2 Power Platform ........................................ 10
2.3 Business analytics .................................... 11
   2.3.1 Power BI .......................................... 11
   2.3.2 Tableau ........................................... 12
2.4 Application development ............................... 12
   2.4.1 Power Apps ....................................... 13
   2.4.2 Outsystems ....................................... 13
2.5 Process automation .................................... 14
   2.5.1 Power Automate ................................... 14
   2.5.2 Zapier ............................................. 15

## 3 Proposed Methodology

3.1 Methodology ........................................... 17
3.2 Work Plan ............................................. 18
3.3 Problem ................................................ 22
3.4 Functional and Non-Functional Requirements ........ 24
3.5 Proposed solution ................................... 25
4 Work

4.1 Setup

4.1.1 System Architecture

4.1.2 Data

4.1.3 Database

4.2 Management of SBU resources

4.2.1 Dashboards

4.2.2 Application

4.2.3 Automated Flows

4.3 Projects Overview

4.3.1 Dashboards

4.3.2 Automated Flows

4.4 Automated processes

4.4.1 Employees projects

4.4.2 Timesheets

4.5 Testing

4.6 Work Analysis

5 Conclusion

5.1 Future Work

A System dashboards

A.1 Employees Allocation Overview with date filters

A.2 Employees Allocation Overview with information filters

A.3 Employees Allocation Overview with matrix filters

A.4 Employees Per Project with information filters

A.5 Employees Per Project with matrix filters

A.6 Employees Abroad with date filters

A.7 Employees Abroad with matrix filters

A.8 Employees Overview KPIs with date filters

A.9 Projects Overview with date filters

A.10 Projects Overview with information filters

A.11 Projects Overview with matrix filters

B System application

B.1 Employees with work model filters

B.2 Employees with SBU filters

B.3 Employee Details edit notes
List of Figures

2.1 Components of Power Platform - source [8] . . . . . . . . . . . . . . . . 10
3.1 Final Planning Tasks . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
3.2 Final Planning Chart . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
3.3 Actual Work Tasks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
3.4 Actual Work Chart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
4.1 System Architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
4.2 Database schema . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
4.3 Employees Allocation Overview . . . . . . . . . . . . . . . . . . . . . . 42
4.4 Employees Per Project . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
4.5 Employees Abroad . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
4.6 Employees Overview KPIs . . . . . . . . . . . . . . . . . . . . . . . . . . 48
4.7 Employees . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
4.8 Employee Details . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
4.9 Employee Abroad Details . . . . . . . . . . . . . . . . . . . . . . . . . . 55
4.10 Office Input . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
4.11 New charge code . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
4.12 Update to a charge code . . . . . . . . . . . . . . . . . . . . . . . . . . 61
4.13 Request new accommodation . . . . . . . . . . . . . . . . . . . . . . . . 62
4.14 Request update to existing accommodation . . . . . . . . . . . . . . . . . 63
4.15 Request cancellation of existing accommodation . . . . . . . . . . . . . . 64
4.16 Request one way flight ticket . . . . . . . . . . . . . . . . . . . . . . . . 65
4.17 Request return flight ticket . . . . . . . . . . . . . . . . . . . . . . . . . 66
4.18 Request update to existing flight ticket . . . . . . . . . . . . . . . . . . . . 67
4.19 Request cancellation of existing flight ticket . . . . . . . . . . . . . . . . . 68
4.20 Employee Charge code reaching end date . . . . . . . . . . . . . . . . . . 70
4.21 Accommodation reaching end date . . . . . . . . . . . . . . . . . . . . . 71
4.22 Flight ticket needed to be booked . . . . . . . . . . . . . . . . . . . . . . 72
4.23 Projects Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 74
4.24 Charge code reaching end date . . . . . . . . . . . . . . . . . . . . . . . 76
A.1 Employees Allocation Overview with date filters . . . . . . . . . . . . . 91
List of Tables

<table>
<thead>
<tr>
<th>Table Number</th>
<th>Table Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Functional Requirements</td>
<td>24</td>
</tr>
<tr>
<td>3.2</td>
<td>Non-Functional Requirements</td>
<td>25</td>
</tr>
<tr>
<td>4.1</td>
<td>Resume table</td>
<td>32</td>
</tr>
<tr>
<td>4.2</td>
<td>Accommodation table</td>
<td>36</td>
</tr>
<tr>
<td>4.3</td>
<td>Flight table</td>
<td>36</td>
</tr>
<tr>
<td>4.4</td>
<td>Employee_Charge_Code table</td>
<td>37</td>
</tr>
<tr>
<td>4.5</td>
<td>Employee_Notes table</td>
<td>37</td>
</tr>
<tr>
<td>4.6</td>
<td>User_Access table</td>
<td>38</td>
</tr>
<tr>
<td>4.7</td>
<td>Employee_Details table</td>
<td>39</td>
</tr>
<tr>
<td>4.8</td>
<td>Skill table</td>
<td>39</td>
</tr>
<tr>
<td>4.9</td>
<td>Language table</td>
<td>40</td>
</tr>
<tr>
<td>4.10</td>
<td>ohrm_project table</td>
<td>40</td>
</tr>
<tr>
<td>4.11</td>
<td>fmr_table</td>
<td>40</td>
</tr>
<tr>
<td>4.12</td>
<td>Opportunity table</td>
<td>41</td>
</tr>
<tr>
<td>4.13</td>
<td>Account_Name table</td>
<td>41</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

In this chapter I will provide context for this thesis. I will start by giving a description of the company, the departments relevant to understand this work and also the hierarchy followed inside the company. After understanding how the company is structured, the problem this thesis is trying to solve will be described and after that a brief description of the solution proposed will be explained. After the context I will describe the motivation for doing this work and also the goals to be achieved. At the end of this chapter the structure of the document is presented.

1.1 Context

InnoWave is an international Information Technology (IT) company, with experience in various areas such as internet of thing (IoT), mobile applications, artificial intelligence, among others. InnoWave has its own products, which were conceived, developed and marketed by it, but it also offers technological consultancy services, where the client indicates what needs he has and InnoWave develops a solution specifically for that client.

Since it is an international company this raises issues on the human resource management of accommodation and dates of the employees working abroad. This thesis tries to solve this and other issues that will be described later, by creating a system where all this information can be centralized and easily updated as needed. Now I will give more details about InnoWave and then describe the existing problems.

InnoWave was founded in 2008 with headquarters in Lisbon and in a space of 12 years it has grown exponentially, currently having offices in Coimbra, Porto, United Kingdom, Belgium, Holland, United States of America and India, employs more than 350 employees and works with more than 100 clients in more than 20 countries.

InnoWave, like all companies, is divided into several departments, of which I will only mention those that will interact directly with the system, the others will be omitted for not being connected to this project.

One of these departments is the Project Management Office (PMO), responsible for
ensuring that the allocation of human resources is in accordance with the company’s needs, it has an overall vision of all the company’s projects so that it is possible to know what the needs are for current or future projects, always depending on the skills that are required on a given project.

Another department is the Office, responsible for renting an apartment for an employee who will work abroad and in addition to this, the employee will also need plane tickets for both the outward and the return trip, which is also done by this department.

And finally, we have the Strategic Business Units (SBU), which represent the development departments of InnoWave, there are 5 SBUs each with its own project expertise. The SBUs that exist are:

- **Enterprise Solutions** - The goal of this SBU is to implement innovative solutions using industry best practices that empower organizations with tools and technology to improve and transform their business. It helps organisations extract more value from their technology investments and improve operational effectiveness, by promoting better alignment of IT systems and business processes.

- **Quality Assurance** - This SBU aims, through good practices and methods, to prove that a certain software fulfills its requirements, depending on the result of the tests done the software is approved or not.

- **IoT** - The focus of this SBU are projects whose components have an IoT aspect or projects that do not have this aspect but want to improve it by using IoT technologies.

- **TV and Media** - SBU focused on software development for devices such as smart TVs.

- **CyberSecurity** - SBU that not only helps with the internal security of the company but also provides security services to clients.

So far it is already known the structure of the company, that is important for this thesis, another relevant topic is the hierarchy of the employees, this is a crucial part of this work due to data restriction being a mandatory requirement, only people who have access to the information should be able to see it, and do not allow other personal to see it.

Each SBU has a common hierarchy:

- **SBU Leader** - Leader of the SBU which has a global view of all the clients and projects of its SBU. The SBU Leader also needs to have into account the revenue generated by those projects and if it is achieving the goals set for the SBU, in order to understand if it is on the correct path to be profitable or if there is a need to change.
• **Account Managers** - Reports directly to the SBU Leader, each account manager is responsible for a part of the SBU clients. It has some of the same responsibilities that an SBU Leader has but it only needs to be concerned about his clients and projects, not all of the SBU clients.

• **Team leaders** - Responsible for several employees and reports to an account manager. The tasks of a team leader is to have meetings with their employees to find out how his motivation in the company is or if there are any pending situations that needs their assistance. They are not exclusively team leaders, in addition to this role they also have positions in clients such as developers, business analysts, testers among others.

• **Employees** - They work either in clients or on internal InnoWave projects. They have the responsibility of developing the actual solution for the client and also need to be in constant improvement of their skills and knowledge.

Now that there is context on how the company is organized, I describe the problem that currently exists, which can be summarized as the decentralization of information.

As an SBU Leader, it is currently difficult and time costly to obtain an overview of its SBU. To know its profitability, clients, current/future projects and employees, it is necessary to interact with different platforms in order to obtain this information and then place it on an excel sheet so that an overview of the SBU can be obtained. Account Managers, they need to keep track of all the clients and projects that are under their care, familiar to what an SBU Leader also needs to do but on a smaller scale.

Projects can either be nearshore, where the employee works remotely for the client, or on site, employee works abroad at the client’s premises. For employees who work abroad it is necessary to arrange accommodation and airplane tickets, information regarding the start and end dates of the accommodation and the flight date is currently only stored in emails, it is up to the Account Manager to save it in order to follow up abroad employees.

Taking into account the described problems the proposed solution will be a system where all this information will be centralized automatically for the user, it will be capable of adding, editing and removing information regarding employees, projects and abroad employees and lastly it will also be capable of sending automatic notification emails to an employee about a new project, requesting accommodations or flight tickets for abroad employees to the Office department or notify managers about close end dates.

### 1.2 Motivation

In any company, independent of its business, it is necessary to have information regarding its current status. If such information is not available it will not be possible to know if the
company is making profit and without this it will be impossible to identify the company problems.

When a company starts to grow it will have more needs, which is the case of InnoWave, and to be able to respond to all its needs, it needs multiple softwares, such as the Orange HRM that is used by the company to fill in the timesheets, the internal curriculum and other points that will be mentioned later. SalesForce is another software where projects and budgets are managed and other functionalities, which will be mentioned in Chapter 2.

These platforms do not exchange information with each other, so it is always necessary to cross information from multiple platforms in order to obtain an overview, this work is done manually by the managers, work that can be automated and improve the managers effectiveness.

This was part of my motivation, being able to develop a system that is able to connect to the various platforms existing in InnoWave, cross this information automatically and show it to the user in a concise way that brings added value.

Another part of my motivation was that in order to be able to plan, design and implement the system, it was necessary for me to understand how the company works, so that the final solution would fit as best as possible the company needs.

1.3 Goals

The goals of the system implemented in this thesis can be summarized on the following points:

- The system should centralize the information automatically for a manager in order for him to have a global view of his SBU and projects in one place.

- In the system it should be possible to add, edit or remove information about an employee or a project.

- The system should send automatic emails notifying managers about important dates.

- The system will need to work with real-time data.

- The information to be shown on the system needs to take into consideration the user.

1.4 Structure of the document

This document is organized as follows:
• Chapter 2 - In this chapter I will present what are the softwares available at InnoWave, the existing tools for data analysis, creating applications and automation of processes and the reasons for choosing the tools used for development in this thesis.

• Chapter 3 - On chapter 3 it will be presented the methodology and tools used during this thesis, the work plan and deviations from the initial planning, the problem will be explained in more detail, the functional and non-functional requirements are specified and then the proposed solution is presented and described.

• Chapter 4 - This is the main chapter of this thesis where all the work will be described, starting with the system architecture and database, followed by the description of all the work including tests done and to conclude the chapter an analysis of the work is done.

• Chapter 5 - Last chapter of this thesis where conclusions are done and future work is mentioned.
Chapter 2

Related Work

This is the chapter where I will first describe the various tools existing in InnoWave that will be linked to my system, then mention the purpose of each of them, what features are available and what information will be used from each one. My system was developed on Microsoft’s Power Platform. I will describe the various components that make it up as well as the reasons why it was chosen and the main competitors on the market will be described.

2.1 InnoWave’s management tools

InnoWave uses several different tools to manage the whole organization, the following will be interconnected with my system: Salesforce CRM, Orange HRM, Finance Management Tool and a new HRM tool.

The personal data about each employee, as well as to which SBU they belong and who their manager is, this information could be found on Lanteria HRM[3] platform that was in place on the company at the beginning of this thesis, but this was changed in the middle of the work to a new one called WeGrow, which is built on the Cornerstone platform[9].

Due to this change, it was decided that only a table with all the necessary information would be created and that when the integration is to be done, the necessary changes would be made, both the information and the table will be described in the HRM table section of the Work chapter.

InnoWave has other platforms for management but those will be out of scope from this thesis since my system will not interact with them.

2.1.1 Salesforce CRM

Salesforce.com’s Customer Relationship Management (CRM)[11] is a technology to manage all the company’s relationships and interactions with customers and prospects, where the goal is to improve business relationships. Salesforce CRM helps companies of all
sizes to accelerate sales, automate tasks and make smarter decisions, it offers many features such as: Contact Management, Sales Opportunity Management, Workflow Rules & Automation, Customizable Reports & Dashboards and a Mobile Application.

The result of implementing the CRM system for the business should be a better customer experience and ultimately higher revenue. The CRM system software could be accessible through a web browser like Salesforce.

The part that will impact my system will be related to project management by managers, where the creation of opportunities will be done, which in the content of this thesis will be referred to as charge code or project, both meaning the same. These serve not only to be able to record work hours on the time sheets existing in the Orange HRM platform, but also to be able to associate eventual costs, which can be done on the Finance platform.

In InnoWave the notion of project is associated with a charge code that can be used in two ways: by a single person, where an employee is assigned a charge code that only he can use, however, this does not mean that he is in a project alone, another way to use the charge code is as a team, where all members use the same charge code.

**Used Information**

In my system I need information related to the charge code such as: the charge code itself, start and end dates, who is its owner and the name of the customer, both the information and the tables will be described in the Salesforce tables section of the Work chapter.

### 2.1.2 Orange HRM

OrangeHRM is an open source HRM solution tailored to the needs of small and medium sized businesses. It is designed to be user-friendly with a low-cost pricing model.

The system is fully customizable and is designed to serve the entire employee lifecycle, from recruitment to employment to leaving the company.

Being open source, OrangeHRM has a large and thriving community of users who contribute ideas and feedback to the ongoing development of the solution. All staff in an organization have access to the system either through its core functions or through an employee self-service portal. A mobile application can also be tailored to individual user needs and access, allowing companies to offer OrangeHRM to any employee with an iOS or Android device.

OrangeHRM is composed of multiple modules: core HR, time and attendance, applicant recruitment, performance appraisals, absence management, benefits administration, recruitment, and more. It provides role-based system access and permission levels for different sets of users.

The Personal Information Management (PIM) module allows companies to capture employee data in a format suitable for auditing and reporting purposes. In addition,
also provides tools for managing employees absences, as well as keeping track of their absence requests. Its functionality is backed by a professional and knowledgeable support.

The modules used from the platform at InnoWave that are used by my system are the following:

1. **Core** - Contains some details about the employees such as name, date of birth and email and it is also in this module that there is the InnoCV which is an internal curriculum vitae of InnoWave where each employee has to update the information regarding which projects he has already worked on, what professional experience, skills and languages he knows plus the various training that he has already done, in addition to this information it is also in this module that the various roles of access to the platform are configured.

2. **Leave Management** - Through this module employees can submit their vacations that will be approved by the manager.

3. **Time & Attendance** - With this module it is possible for each employee to enter the hours he worked during the week on the timesheet, associating these to a charge code and that after being submitted, the manager can approve it.

**Used Information**

In my implemented system I needed information related to each employee’s skills, languages they know, timesheets and active charge codes, both the information and the tables will be described in the Orange tables section of the Work chapter.

### 2.1.3 Finance Department Tool

This is a tool that has been developed within InnoWave for internal use, where employees can submit expenses, which are either approved or not by the manager. Here the Office department can enter expenses for airline tickets or accommodations of employees working abroad.

Each expense submitted on this platform is called a Purchase Order (PO) which has a mandatory charge code, expense description and an id that is automatically generated when it is created.

**Used Information**

My system will not interact with this platform, it will only contain the PO IDs generated, which will also be entered by the Office on my system.
2.2 Power Platform

Power Platform[7] is a system from Microsoft that allows its users to do three key actions for the realization of their business: Analyze, Act and Automate. This is done through Power BI, Power Apps and Power Automate (previously called Power Flow). Power Virtual Agents is also part of the Power Platform, which allows the easy creation of personalized chat bots, but since this component is not of interest for this thesis, it will not be discriminated.

The Power Platform was created to give an advantage to people who are not familiar with programming or who are not experts in data analysis software, but who want to increase the productivity of their businesses. Using this platform they can easily create dashboards, get an overview of a company’s business and apply filters to view specific information, create applications to be used by employees, and create automatic processes that are put into action with a single click or run automatically.

The reasons I choose Power Platform to do my thesis development were as follows:

1. Since it is a low-code/no-code platform, the complexity of use associated with it is low, so it is possible for a person with little or no programming knowledge to develop a solution with a small learning curve.

2. Currently in the company there are already dashboards created in Power BI, Power Apps applications and automatic flows in Power Automate, so the platform is already known by managers so it will be easy to use my system.

3. InnoWave has a partnership with Microsoft and much of the software used is Microsoft software, such as SharePoint, Office, Project and other existing software,
which can be linked together with relative ease. Each of the components of the Power Platform are widely used in companies due to the ease of use and benefits they offer, all of them can be used individually but it is always preferable to use the various components simultaneously in order to get the most out of the platform.

4. All components are prepared to communicate with each other easily - Since Power BI, Power Apps and Power Automate together with Power Virtual Agents form the Power Platform, they all have pre-defined connectors for easy interconnection between all of them, but it also has many other connectors to known existing systems.

2.3 Business analytics

In this section I describe both Power BI and Tableau Desktop which are competitive business analytics tools.

2.3.1 Power BI

Power BI is a component that makes it easy to connect, analyze and understand the data that matters to the business, wherever that data is located: in the cloud, in an Excel table, in a SharePoint list, in an Oracle, SQL or Salesforce database, or in any of the hundreds of systems supported by the Power Platform.

Power BI was the first Power Platform service to be released to the market, in January 2015. The desktop version of the software is available for free, but the biggest drawback is when trying to access cloud services. The desktop version needs a work account to log in, a public account will not work here, and depending on the service we want to connect to a premium account may be necessary, otherwise a free account will not have access to these premium connectors.

Microsoft Power BI supports two different languages, M and Data Analysis Expression (DAX) that can be used to filter, manage and visualize data. M can be thought of as a query formula language and can be used in the Power BI Query Editor in order to prepare data before it is loaded into the Power BI model. On the other hand, DAX is an analytical data calculation language that can be used for deep data analysis during the Data Visualization phase.

M and DAX are not dependent on each other and follow completely different structures and logic, and have different underlying code. M and DAX cannot be used simultaneously, since the M language is used in the Query Editor, while DAX is mainly used in the Data View model.

Power BI is a very flexible data visualization tool that allows analysts to create custom calculations to perform data analysis. These calculations are mainly Columns or Calculated Measures, both of which are defined in DAX.
Columns are usually referred to with the following syntax 'TableName'[ColumnName]. These columns can be taken from the original data source or created in Power BI with a DAX expression.

Measures are referred to by their name in square brackets [MeasureName]. Measures are best thought of as dynamic aggregations of values. Their value will ultimately depend on the level on which they are calculated. They are table independent and offer developers many analytical possibilities.

The main difference between calculated columns and measures is that columns are evaluated on every row, while measures are evaluated only at the level of granularity at which they are created. Columns are calculated when the data is updated, while measures are recalculated at each visual interaction. This means that, unlike measures, column values do not respond to the selection of filters or parameters.

To work with Power BI no programming knowledge is required, which allows its use by all people, it is capable of processing a large amount of data, but this will be useless if all this information is not updated. For this purpose, there are Power Apps and Power Automate.

2.3.2 Tableau

Tableau is an integrated Business Intelligence (BI) and analytics solution that helps analyze key business data and generate meaningful insights. The solution helps businesses collect data from multiple points of origin such as SQL databases, spreadsheets, cloud applications like Google Analytics and Salesforce to create a collective data set.

The user interface is intuitive, a wide variety of charts are available. For simple calculations and statistics, no coding skills are required, but for heavy analysis, models can be run in R, a programming language for analytics, and then import the results into Tableau, which requires quite a bit of programming skill based on the task needed to perform.

Tableau has a free public version at no cost, but provides only 1GB of online storage, for the server and desktop version a paid license is required for each.

Tableau’s live visual analytics and interactive dashboard allows its users to generate relevant insights and explore new opportunities. Users can create interactive maps and analyze data across regions, territories, demographics and more. Users can also view and analyze data on their Android and iOS smartphones with the Tableau Mobile app.

2.4 Application development

In this section I describe both Power Apps and Outsystems which are competitive low-code application development tools.
2.4.1 Power Apps

Power Apps\[4\] allows anyone without application development skills to create an application, supports drag-and-drop installations, and provides users with a collection of templates that allows reuse of already developed applications. A user can follow canvas or model-driven approaches while building applications.

The Power Apps canvas application provides a blank canvas onto which the user can drag and drop components in any formation to design a user interface. Once everything is in place, additional adjustments can be made to the size and formatting of each of these components.

In comparison to canvas applications, model-driven applications in Power Apps are based on underlying data-specifically, data stored in Dataverse. Dataverse is a secure, cloud-based storage space that organizations can use to store business application data.

Since model-driven applications take a data-first approach, they are much more rigid in design than canvas applications, most user interface components are pre-made and pre-selected based on the chosen data, although there is still room to edit individual components. However, this data-first approach also means that model-driven applications are more sophisticated than canvas applications, making them better suited for applications that require complex business logic.

Power Apps integrates with many services in the Microsoft ecosystem, such as Excel, Azure database or legacy systems, it also supports Office 365 integration, pre-built templates, easy conversion of mobile and tablet applications and the ability to link with third-party applications for basic application development.

The natural connection between Power BI and Power Apps makes the integration of a Power App into a company where Power BI dashboards already exist very intuitive and without much effort, the application will always be tailored to each company’s needs, which will increase the company’s productivity.

Like Power BI, Power Apps connect to hundreds of systems and databases, which makes it easy for users to access the existing processes and data that make the company run. The data collected in Power Apps can be used by Power BI, which creates a cycle of continuous improvement.

2.4.2 Outsystems

Outsystems\[10\] is a low-code, AI-powered application development solution designed to modernize the entire application lifecycle, from initial build to cross-platform deployment. Outsystems is a full-function visual development platform that allows users to create and customize user interfaces, business processes, logic and data models, and more with a built-in drag-and-drop application builder. When ready, Outsystems will handle all deployment processes and automatically check for dependencies with a 1-click system.
OutSystems has two significant components:

- It has an intermediate studio for connecting to databases through .NET or Java, and secondly, it has a service studio for specifying the behavior of the application being developed. Some of the applications supported on this platform are billing systems, CRMs, operational dashboards, and business intelligence.

- It also provides developers with a fast mechanism to publish developed applications, the ability to connect different services, to develop responsive mobile devices web-apps, security mechanisms and real-time dashboards.

Outsystem can scale to any enterprise environment, supporting load balancing and auto-scaling capabilities. For collaborative development, OutSystems provides in-system feedback functionality to help integrated teams solve business challenges without any communication barriers.

2.5 Process automation

In this section I describe both Power Automate and Zapier which are competitive process automation tools.

2.5.1 Power Automate

Power Automate is at its core a Robotic Automation Process (RPA), which is the automation of simple tasks, RPA solutions promise to reduce costs and errors. However, Power Automate is an automation intelligence process that goes further than RPA, as it allows complex automation services to be performed by those without extensive IT knowledge.

In Power Automate all flows are step-based, and in each step the user can choose from a wide list of supported actions, they can be either to apply logic to model information or to connect to supported external services.

There are multiple types of flows in Power Automate which are described as follows:

1. **Cloud flow** - Flows used when an automation is to be triggered automatically, instantly, or through a schedule.

2. **Desktop flow** - Flows to automate tasks both on desktop and web.

3. **Business process flow** - Flows to help employees follow the same steps for a specific task or process.

Through a simple and intuitive interface, the user can create an automated workflow, triggered from Power BI data, that can be edited in Power Apps and integrated with any of the systems supported by the Power Platform.
2.5.2 Zapier

Zapier[14] is a cloud-based integration platform that helps companies of all sizes automate daily workflows involving multiple applications. The solution allows users to automatically transition data between web applications while integrating applications with one-click functionality. Zapier allows users to define actions and set automatic triggers.

Key features of Zapier include automatic reminders for repetitive tasks, online templates, activity dashboard, API maintenance, triggers and actions, online development portal, new application updates and account management.

Zapier integrates with applications such as Slack, Google Sheets, QuickBooks, Facebook Lead Ads, Google Docs and more. Support is provided through documentation and an online help desk. Pricing is based on a monthly and annual subscription.
Chapter 3

Proposed Methodology

In this chapter, the work methodology for carrying out this thesis will be presented as well as the work plan, taking into consideration the initial planning. Once the outline of the work is done, the problem is described in more detail after that the functional and non-functional requirements are specified taking into consideration the problem, then the proposed solution is described and finally the used tools for the development of my system are detailed.

3.1 Methodology

The work organisation for this thesis was based on SCRUM[12], which is a methodology widely used in software development projects due to its versatility of being able to introduce changes in a product even if the development has already started.

To take full advantage of this framework, some roles exist in order to effectively divide the work in the team:

1. **Product Owner** - Person who knows the business and has to find the balance between what the development team is able to deliver and what is needed.

2. **Scrum Master** - Organizes all the meetings, unblocks situations for the developers and also organizes tasks for the developers.

3. **Development Team** - Implement the product itself.

   My team was composed by 3 people, me, Tiago Correia and Daniel Pinto. Where my roles on the team were both the developer and scrum master, I had to implement the system and based on the requirements organize the tasks. Both Tiago and Daniel played the role of Product Owner where their job was to give guidance on what was needed and best to be implemented.

   In SCRUM there are the so-called Sprints, they are a time frame defined by the team in which it defines what the goal will be in that time and consequently the increment to the product that will be made.
Sprints can have a different duration depending on the team and the software to be developed. 2 weeks is the recommended time, but for my thesis it was decided by the team that the duration would be 3 weeks due to the fact that I was the only developer, 2 weeks was considered a short time to have something implemented and ready to show at the end of the Sprint.

The work plan of my thesis was divided into 7 tasks, which were completed with several sprints each, which totaled 13. Both tasks and sprints themselves will be described in the next topic where I will describe the work plan and provide reasons for some deviations to the initial plan and finally the work that was actually done.

### 3.2 Work Plan

The theme of this thesis was created based on meetings with all the SBU Leaders in order to understand their difficulties in their daily work and on their SBU.

The initial work was defined after all the meetings and a final meeting with Tiago and Daniel, the plan consisted in the development of 2 modules and the planning of another 2 those being the following respectively.

1. **Resource Management (Team Leaders)** - Centralizing the information regarding the employees of each manager in an easily accessible and useful manner.

2. **New Employee Request** - Allowing the platform to request an employee for a project.

3. **Resource Management (Human Resources)** - This module will also centralize information regarding employees like the Module 1 but this one will be used by the PMO and not the managers of an SBU. This module will contain different information because the needs of the departments are not the same.

4. **Bilability** - Financial details of an SBU such as profit.

The modules mentioned above were defined in May 2020 but the thesis only started in October 2020, a difference of 5 months, which in an innovation company that always tries to improve and optimize its processes means a lot of time.

This means the work plan had to be updated as most of the planned modules had already been implemented by October and there was a new tool in the company whose functionality would solve the problem I intended to solve in one of the modules.

Module 1 remained with the same scope because currently it does not exist in the company and there was no forecast of anything to be done soon to tackle this, only the module name was modified to be more succinct: Resource Management of an SBU.
Module 2 no longer made sense to be implemented in this thesis due to the fact that the company acquired a new software with the features I wanted to implement, among many others.

Module 3 dashboards were already developed by the time this thesis started, so there was no need to repeat work, the same scenario happened with the Module 4.

Taking into consideration the previous topics the final planning was defined as it follow:

<table>
<thead>
<tr>
<th>Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
<th>Predecessors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 State of the art</td>
<td>30 days</td>
<td>10/12/2020</td>
<td>11/20/2020</td>
<td></td>
</tr>
<tr>
<td>2 Analysis and planning of the module “Resource Management of an SBU”</td>
<td>30 days</td>
<td>11/23/2020</td>
<td>01/01/2021</td>
<td>1</td>
</tr>
<tr>
<td>3 Module implementation</td>
<td>40 days</td>
<td>01/04/2021</td>
<td>02/05/2021</td>
<td>2</td>
</tr>
<tr>
<td>4 Presentation and demonstration of the module</td>
<td>5 days</td>
<td>03/08/2021</td>
<td>03/11/2021</td>
<td>3</td>
</tr>
<tr>
<td>5 Analysis and planning of the module “Projects Overview”</td>
<td>15 days</td>
<td>03/15/2021</td>
<td>04/02/2021</td>
<td>4</td>
</tr>
<tr>
<td>6 Module implementation</td>
<td>25 days</td>
<td>04/05/2021</td>
<td>05/07/2021</td>
<td>5</td>
</tr>
<tr>
<td>7 Presentation and demonstration of the module</td>
<td>5 days</td>
<td>05/10/2021</td>
<td>05/14/2021</td>
<td>6</td>
</tr>
<tr>
<td>8 Analysis of processes that can be automated</td>
<td>10 days</td>
<td>05/17/2021</td>
<td>05/28/2021</td>
<td>7</td>
</tr>
</tbody>
</table>

Figure 3.1: Final Planning Tasks

![Final Planning Chart](image)

Figure 3.2: Final Planning Chart

The final modules to be implemented can be enumerated as follows:

1. **Resource Management of an SBU** - Same focus as the module “Resource Management (Team Leaders)”.

2. **Projects Overview** - Provide managers information about their projects in a succinct and effective way.

3. **Automated Processes** - Automate my platform so that even without manual editing it can stay as close to reality as possible and secondly investigate current processes in the company that can be automated and as a result save the company time and money.

All the tasks of the final planning are detailed in the following list.

- **Task 1** - Investigate about which platforms and technologies are used in the company related to this work and to research platforms that will be used for development.
• **Task 2** - Consists on the analysis and planning of Module 1.

• **Task 3** - Consists on the implementation of Module 1.

• **Task 4** - Presentation regarding Module 1 implementation.

• **Task 5** - Consists on the analysis and planning of Module 2.

• **Task 6** - Consists on the implementation of Module 2.

• **Task 7** - Presentation regarding Module 2 implementation.

• **Task 8** - Consists on the analysis and planning of Module 3.

Having described the final plan of the thesis, I now go on to show the work that was actually done and describe the reasons for some deviations from planning.

As can be seen by comparing both the planning and work done, the plan was followed as expected on the first 2 tasks, but there were some changes on the following ones. In Task 3 there was an initial estimation of 45 days which is equivalent to 3 sprints, but the work that was done lasted 4 sprints.

This was due to the fact that throughout the analysis in Task 2, several new features emerged that could be implemented in the tool, which would bring added value, so it was decided to add one more sprint in this task in order to implement these.

In the initial and final plan, presentations were planned at the end of each module, in order to show the work done and get feedback from the SBU Leaders, but together with Tiago and Daniel it was decided that it was more advantageous to make just one presentation that resulted in Task 6 which occurred on May 27, 2021.
This was attended by the CEO Tiago Gonçalves, 3 SBU Leaders and also the person responsible for the company’s finance department, 2 of the SBU Leaders could not attend due to overlapping meetings so the presentation was recorded allowing all stakeholders to review it in the future.

Since I would only have a final presentation, module 1 and 2 were planned and implemented sequentially as we can see in Tasks 2, 3, 4 and 5 but Module 3 was only completed after the presentation, and since it had no visual or functional impact on the platform it did not prevent me from presenting the first two modules and getting feedback.

The final work can be resumed in the following topics were the tasks are enumerated and the sprints are specified for each one:

**Task 1 - State of the art**

- *Sprint 1* - Investigate about existing platforms in the company and technologies used.

- *Sprint 2* - Research on what tools to use for the work development.

**Task 2 - Analysis and planning of the module ”Resource Management of an SBU”**

- *Sprint 3* - Gather of requirements by having meetings with managers to know in more details their needs.

- *Sprint 4* - Planning of Module 1 where I had to plan the dashboards, the application and the module’s automatic flows.

**Task 3 - Module Implementation**

- *Sprint 5* - Dashboard development.

- *Sprint 6* - Completion of dashboard development and start of application development.

- *Sprint 7* - Application development completion.

- *Sprint 8* - Development of automatic flows.

**Task 4 - Analysis and planning of the module ”Projects Overview”**

- *Sprint 9* - Gather of requirements by having meetings with managers and Module 2 planning in which I had to plan the dashboards and the module’s automatic flow.
Task 5 - Module Implementation

- *Sprint 10* - Dashboard development.
- *Sprint 11* - Completion of the dashboard and automatic flow development.

Task 6 - Presentation and demonstration of all the work so far

- Presentation held on May 27th with the presence of the CEO Tiago Gonçalves and the SBU Leaders.

Task 7 - Analysis of processes that can be automated

- *Sprint 12* - Implementation of autonomous information update on the system.
- *Sprint 13* - Investigation on how currently timesheets are partially pre-filled and plan a solution to fully pre-filled them.

3.3 Problem

In the meetings I had with the SBU Leaders in order to define the scope of the thesis, several problems were listed and based on these problems, the modules to plan and implement were defined. These meetings took place in May 2020, before the thesis proposal was submitted at the faculty and at the time the problems that were mentioned are described below as well as the modules in which they would be solved.

The main focus of this thesis is to solve the decentralization of information regarding employees, thus the first and most important module of this thesis is "Resource Management (Team Leaders)", which underwent a name change in the final planning in order to be more intuitive and that was "Resource Management of an SBU".

Information regarding which charge code is being used by a certain employee, the start and end date of it, date on which an employee will be without a project, know their current project and if any, future project, in which projects the employee has already worked as well as the client, their skills and who is the employee’s direct manager, to obtain this information it is necessary to access different platforms and then add this information in an excel sheet, just so it is possible to obtain an overview in one place.

In addition to this information, there is also the employees working abroad, this means that the employee will work on the clients premises where he will need an accommodation and plane tickets. This information only exists in emails sent by the SBU to the Office, unless the manager also saves it somewhere else.

Another important part is the end dates, these can be from a charge code in which the manager, when the end date approaches, has to create a new charge code, another one
is the date referring the end of an employee’s contract with a client which needs to be renewed or allocated to a new project.

For employees abroad the situation is identical, the managers have to know when the accommodation contract is ending, if it will be renewed or if the employee will return and in that case he will need a plane ticket. A way the managers did to get around the problem was to create manual alerts on their calendars and by doing that they would know when the end date was near, which again is a manual job and being manual, some part can be forgotten which is not the perfect scenario.

At the time of my initial meetings with the SBU Leaders, in order to make a requirement of an employee a meeting with several managers was held and an excel sheet was updated with the current needs of the company as well as the people who are about to join or who will be available. Manual editing of an excel sheet means that it is not a process that is optimized or automated, that is why I proposed the “New Resource Request” module, but as already mentioned, InnoWave acquired a tool where it will be possible to apply for a collaborator, know which people could become future employees and in what state of the recruitment process they are, so this module no longer applies.

On the module “Resource Management (Resource humans)” the situation was identical to the previous one. For the PMO department to be able to obtain a global view of the company’s employees, there was an excel sheet that was manually updated and the company’s overview was obtained, but by the time I started the work, dashboards with the required information were already developed, hence this module was also removed from the planning.

The “Bilability” module, where its purpose was to show the financial status of each SBU, was also removed due to the fact that Power BI dashboards were created with this information. Previously the problem that existed was that the company has a platform where it is possible to do and see all the financial part, but the way the data is shown is not user-friendly at all, the managers had to extract the financial reports that the platform provided, then process this data manually and show it in a more succinct way, where Excel was sometimes used again.

During the meetings, the information available to the managers was just about all their projects, past current or future, information that is provided by Salesforce, the issue is that in Salesforce, a charge code created for a project is not associated with a person, some managers try to get around this problem by indicating the name of the person on the charge code itself, but this process cannot be applied to all cases, as there are charge codes that are used by more than one employee. I solve this problem on the first module where I will save a connection between an employee and a charge code.

Another need the managers have is to know what their current and future projects are and what people are working on them. This is why I proposed the module “Projects Overview” where I try to solve this problem. Furthermore Salesforce does not notify the
manager about their projects being close to completion which again requires managers to manually create reminders.

Finally, there is one global problem in the company which is filling in the timesheets, despite being a relatively simple task, many employees do not fill theirs in time, which eventually causes problems for the company, because the timesheets are a very important process in order to pay the employees salaries in time. Due to the existence of this problem I decided to try and find a solution that will pre-fill them as much as possible for each employee automatically.

3.4 Functional and Non-Functional Requirements

Based on the described problems the following functional requirements were specified:

<table>
<thead>
<tr>
<th>Requirement #</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement 1</td>
<td>Show the allocation of an employee within a user-specified time frame.</td>
</tr>
<tr>
<td>Requirement 2</td>
<td>Show each employee’s project history.</td>
</tr>
<tr>
<td>Requirement 3</td>
<td>Show the end date of an employee on a project.</td>
</tr>
<tr>
<td>Requirement 4</td>
<td>Show employees abroad as well as information about their accommodation and airplane tickets.</td>
</tr>
<tr>
<td>Requirement 5</td>
<td>Show the following summary numbers of an SBU: Employees without a project, Partially allocated employees, Fully allocated employees, Employees close to being without a project, Employees whose accommodation is close to the end date, Airplane tickets required to request the Office and Total number of employees the user has access to as well as the total number of employees abroad.</td>
</tr>
<tr>
<td>Requirement 6</td>
<td>Show the following information about charge codes: End date, Customer name and Employees using them.</td>
</tr>
<tr>
<td>Requirement 7</td>
<td>Show the total number of clients and projects within a user-specified time frame.</td>
</tr>
<tr>
<td>Requirement 8</td>
<td>Add, edit and remove the association of an employee to a project, where the charge code, work model, start date, end date and allocation are specified.</td>
</tr>
<tr>
<td>Requirement 9</td>
<td>Add, edit and remove an accommodation request to the Office, where the charge code, location, start date, end date and notes if needed, are specified.</td>
</tr>
<tr>
<td>Requirement 10</td>
<td>Add, edit and remove the request for a flight ticket to the Office, where the charge code, city of departure, city of arrival, minimum flight date and time and notes if needed, are specified.</td>
</tr>
<tr>
<td>Requirement 11</td>
<td>Notify an employee of its new charge code and on what dates he should use it, as well as any changes made to its current charge code.</td>
</tr>
<tr>
<td>Requirement 12</td>
<td>Notify managers about upcoming end dates for an employee on a project, accommodation, plane ticket and charge code.</td>
</tr>
</tbody>
</table>

Table 3.1: Functional Requirements

The following non-functional requirements were specified for the system that would be developed:
Chapter 3. Proposed Methodology

| Requirement 1 | The system should run on any platform. |
| Requirement 2 | System response times must not be long, it must be a responsive system. |
| Requirement 3 | The system interface must be user-friendly. |
| Requirement 4 | The information shown must take the user into account. |
| Requirement 5 | The system must support several different users at the same time. |
| Requirement 6 | Time to learn how to use the system must be short. |
| Requirement 7 | The system must be built in such a way that it will be relatively easy to add new modules in the future. |
| Requirement 8 | Easy implementation of new modules by other employees. |
| Requirement 9 | The system must be developed in a sandbox environment. |

Table 3.2: Non-Functional Requirements

3.5 Proposed solution

Taking into account the problems described previously now I will make a high-level description of the implemented solution, the detailed description will be done in the chapter 4 chapter.

The first module is not only the most important, it is also the module where more features were implemented, to be more precise, 4 dashboards were created using Power BI:

1. **Employees Allocation Overview** - Presents the entire history as well as the future of each employee in terms of allocation in projects, it will be possible to see, at a given time, what was the percentage of allocation of an employee, allowing the manager to easily know what employees he has available on certain periods of time.

2. **Employees Per Project** - Possible to see the entire history of each employee in relation to the projects where he has been allocated, information such as charge code, allocation, start date, end date, work model and other information will be provided, this way, managers will have a dashboard where it will be possible to know which charge codes an employee has already been allocated to.

3. **Employees Abroad** - Managers will be able see, broken down in a matrix, when employees will be abroad. This is due to the fact that it is shown in which days an employee has an accommodation booked. Information such as the location, when the contract end date will be near the current date, if there is already a purchased flight ticket for both the outbound and the return trip will also be shown here.

4. **Employees Overview KPIs** - Dashboard where important values of an SBU are shown, such as the number of employees who are currently not fully allocated to a project.
An application was also created in the first module using Power Apps, where it is possible to see the same information visible on the dashboards but here it is possible to update it. For this application 4 screens were created:

1. **Employees** - Contains list of all employees that the connected manager is allowed to see, it will be possible to filter the list based on the SBU it belongs or by the employee’s name.

2. **Employee Details** - Details of an employee are presented, for example, their professional summary, the projects where they have been, are or will be in, their skills and languages and their level of knowledge and it is also on this screen that it is possible to allocate a new charge code to an employee, edit an existing one and also navigate to the “Employee Abroad Details” screen.

3. **Employee Abroad Details** - On this screen, each manager can see the accommodations that a certain employee has already been on, as well as the flight tickets, in addition to being able to see this information, the manager can edit or make a new accommodation or flight ticket request.

4. **Office Input** - The data entered on this screen will only be one or two Id’s, this id comes from a PO which will be created by the Office taking into consideration the request made by the SBU, which in the context of my thesis will only be about either accommodations or airline tickets. This Id allows the manager, when he consults the “Employee Abroad Details” screen, to know which PO is associated with the accommodation or flight ticket request.

The last part of this module is the automation of processes developed with the help of Power Automate and which can be divided into 2 groups. In the first group there are only automatic email flows depending on the manager input in the application, there are in total 8 different emails in this first group.

1. **New charge code** - Email sent to an employee informing him of the new charge code he should use and on what dates.

2. **Update to a charge code** - Email sent an employee informing him of updates made to a charge code associated with him.

3. **New accommodation** - Email sent to the Office requesting an accommodation for an employee where information such as the city, dates, associated charge code, to associate costs in the PO, and extra information that the manager deems necessary for the Office.

4. **Update to existing accommodation** - Email sent to the Office informing changes that need to be made to an accommodation that has already been requested.
5. **Cancellation of existing accommodation** - Email sent to the Office informing that the accommodation will no longer take effect and therefore can be canceled if it has already been booked.

6. **New flight ticket** - Email sent to the Office requesting a flight ticket for an employee where information such as departure and arrival city, minimum flight date and time, associated charge code, to associate costs in the PO, and extra information that the manager deems necessary for the Office.

7. **Update to existing flight ticket** - Email sent to the Office informing changes that need to be made to a flight ticket that has already been requested.

8. **Cancellation of existing flight ticket** - Email sent to the Office informing that the flight ticket will no longer take effect and therefore can be canceled if it has already been booked.

The second group is composed of alert emails, about end dates, that are automatically sent to the managers depending on the existing data in the database, these being:

1. **Employee charge code reaching end date** - Contains information about the charge code of an employee being close to the end date and there is still no project planned for the employee.

2. **Accommodation reaching end date** - Information regarding accommodations of employees abroad whose contract is close to expiring and there is still no scheduled return plane ticket.

3. **Flight ticket needed to be booked** - Email sent when an accommodation is already booked, but there is no outbound flight booked yet.

In the second module "Projects Overview" one dashboard and one automated flow were be developed. The dashboard being the following:

1. **Projects Overview** - In this dashboard it will be possible for managers to see which projects exist and which employees are involved, as well as their end date and values such as the total number of projects, clients and employees working on the projects are also shown.

The automated flow of this module is:

1. **Charge code reaching end date** - The automatic flow created in this module consists of sending an email to a project manager to alert that a certain charge code is close to its end date.

Finally, there is the third and last module of this thesis, which is composed by 2 tasks:
Task 1 - Taking into account the timesheets submitted by each employee, will allow my platform to have autonomy to update the data related to the projects in which each employee is automatically, without the need for a manager to edit the information manually, but even so this can be edited by the manager because sometimes the information might differ from reality.

Task 2 - Present a possible solution on how to implement the automation of filling timesheets, this may be possible due to my table where there is data on which charge code an employee is supposed to be using on certain dates.

3.6 Used Tools

In this section I describe all the languages, tools, frameworks and software that were used throughout the work, thus showing my working environment.

Languages

Four different programming languages were used which are as follows:

1. **Java** - Java is an object-oriented programming language that produces software for multiple platforms. The compiled code from a Java application runs on most operating systems (OS), including Windows, Linux and Mac OS. Java derives much of its syntax from the C and C++ programming languages.

2. **DAX** - DAX stands for Data Analysis Expressions, it is a language developed by Microsoft to interact with data in a variety of their platforms like Power BI and PowerPivot. It is designed to be simple and easy to learn while exposing the power and flexibility of tabular models, it can be compared with Excel formulas but more powerful.

3. **Power Fx** - It is the language that is at the heart of Microsoft Power Apps canvas apps today and is inspired by Microsoft Excel. Power Fx describes business logic in concise, yet powerful, formulas. Most logic can be reduced to one-liners with plenty of expressiveness and control for more complex needs.

4. **SQL** - SQL stands for Structured Query Language, it is a domain-specific language. It was used during the application development to enable a programmer to work with data. The data is stored in a relational database. To manage this data there are relational database management systems like SQL Server, MySQL, among others.
Tool and Frameworks

One tool and two different frameworks were used which are as follows:

1. **Maven** - Maven is a powerful project management tool that is based on Project Object Model (POM). It is used for projects build, dependencies and documentation. In short terms maven is a tool that can be used for building and managing any Java-based project.

2. **Spring** - Spring is the most popular application development framework for enterprise Java. Spring framework is an open source Java platform. The core features of the Spring Framework can be used in developing any Java application and there are extensions for building web applications.

3. **JUnit** - JUnit is a Java unit testing framework, one of the best test methods for regression testing. An open-source framework, it is used to write and run repeatable automated tests.

Software

Five different software programs were used, which are as follows:

1. **Microsoft Office** - Microsoft Office is a suite of applications designed to help with productivity and completing common tasks on a computer. A person can create and edit documents containing text and images, work with data in spreadsheets and databases, and create presentations and posters. On this thesis only Excel and PowerPoint were used.

2. **Microsoft Power Platform** - The Microsoft Power Platform is a powerful set of applications that allow the automation of processes, build solutions, analyze data, and create virtual agents. In this thesis Power Virtual Agents was not used.

3. **Microsoft SQL Server Management Studio** - SQL Server Management Studio is a management tool from Microsoft that provides a graphic interface for working with SQL Server database servers.

4. **Microsoft Teams** - Microsoft Teams is a chat and collaboration platform where all conversations, meetings, files, and notes can be accessed by team members in one place that is part of the Office 365 suite of services.

5. **IntelliJ IDEA** - IntelliJ IDEA is an Integrated Development Environment (IDE) for Java language created by JetBrains, used primarily for developing programs for the Java Virtual Machine (JVM).
The reasons for using Power Platform have already been mentioned in the section Power Platform of chapter Related Work and the reason for using Teams is because it is the internal communication tool in the company.

The criteria taken into account when choosing the tools for the development of Task 1 of the last module of this thesis were:

1. **Prior knowledge of the technologies** - At Innowave I am a backend Java developer, most of the tools I use on a daily basis are Maven, Spring, JUnit and IntelliJ, and due to this fact I have a lot of knowledge about these tools which was an asset for the development of this thesis.

2. **Versatility** - Java code, as already mentioned, is capable of running on most operating systems, and for this reason it is easier to have a server capable of running the program on.

3. **Community and Support** - All the technologies used have large and reliable communities which helped a lot in the development of the thesis.
Chapter 4

Work

In this chapter, all the work that was done will be described, starting with the architecture of the system and database. Then the 3 modules of this thesis are described, the multiple tests carried out are shown and finally an analysis of the work done is made comparing the solution with the functional and non-functional requirements.

4.1 Setup

This section is composed firstly by the system architecture as well as all the content developed in the 3 modules summarized in a table. The data I worked with is described as well as the restrictions made to it, both in Power BI and Power Apps, and at the end the database is described.

4.1.1 System Architecture

The following image shows the system architecture, it is possible to see that it is composed of 4 entities namely: Database, Power BI where the dashboards are, Power Apps from where the application was created and Power Automate where the various email flows were created.

![Figure 4.1: System Architecture](image)
All elements of the Power Platform communicate with the database through connectors that are already built-in, it is only necessary to indicate the access details. Not counting Power Automate, the other two entities of the system write to the database, Power BI writes information into the User Access table which will be described later on and how this writing happens, Power App writes to several tables with information regarding the allocation of an employee in a project, accommodation or flight ticket requisitions.

It is also possible to see that Power Apps sends information to Power Automate, this is due to the fact that most of the developed flows are triggered from the manager’s actions in the application.

All the features developed in this thesis are summarized in the following table, these can be dashboards, screens in the application, automatic flows of sending emails to an employee or to the Office department as well as emails warning about upcoming end dates sent to managers, Java application to automatically update data and a possible solution for automating the pre-fill of timesheets in the company.

<table>
<thead>
<tr>
<th>Module</th>
<th>Tool</th>
<th>Functionality</th>
</tr>
</thead>
</table>
| Management of SBU resources | Power BI      | Employees Allocation Overview  
Employees Per Project  
Employees Abroad  
Employees Overview KPIs |
|                             | Power Apps    | Employees  
Employee Details  
Employee Abroad Details  
Office Input |
|                             | Power Automate| New charge code  
Update to a charge code  
Request a new accommodation  
Request update for an existing accommodation  
Request cancellation for an existing accommodation  
Request a one way flight ticket  
Request a return flight ticket  
Request update for an existing flight ticket  
Request cancellation for existing flight ticket  
Employee charge code reaching end date  
Accommodation reaching end date  
Flight ticket needed to be booked |
| Projects Overview           | Power BI      | Projects Overview |
|                             | Power Automate| Charge code reaching end date |
| Automated processes         | Java application | Employees projects |
|                             | Planner       | Timesheets |

Table 4.1: Resume table
4.1.2 Data

All data in the database is mocked data, it is based on real data but have been changed so that there are no confidentiality issues. Tiago provided me excel sheets containing existing information from both the Orange HRM database as well as Salesforce data that was extracted from Power BI.

The data I worked with were as follows:

- 4 SBUs - ES, QA, IoT e TV&Media
- 8 Clients - Sonae, Vodafone, WOW, Galp, Proximus, EDP, Altice e SIBS
- 45 Employees
  - 1 CEO
  - 4 SBU Leaders
  - 8 Account Managers
  - 8 Senior Developers
  - 2 Senior Business Analysts
  - 2 Senior Testers
  - 15 Developers
  - 2 Business Analysts
  - 3 Testers

The charge codes created took into account 2 different aspects: charge code duration, which varies from 3 months to 1 year, and charge codes with the employee name which will be exclusively used by him or shared charge codes where several employees use the same.

Data Restriction

An important feature of this platform is that each manager only sees the data they have access to, the rest is filtered out automatically. The rules applied in data filtering are identical in both Power BI and Power Apps, but since they use different programming languages the logic done in each one was a bit different and will be described below.

Power BI  I used the role functionality provided by Power BI, where I created one role that filters only the Employee_Details table. Using the DAX language, I can get the id of the connected manager through their email and by doing a search on Employee_Details table it allows me to know what their SBU is. Using the manager id again, I search for his hierarchical level in the company, which is easily obtained in the User_Access table.
Having this initial information regarding the connected manager, two rules need to be applied namely: an employee is not filtered out if he belongs to the same SBU as the connected manager and is on a hierarchical level below him, another scenario is the employee does not belong to the SBU of the connected manager but is associated with a project of the same SBU, this project is only taken into account if it is currently being executed or will be executed in the future.

With this filtering on the Employee_Details table, only the employees to which the connected manager has access will be shown, it also means that only the data that is related to each employee will be shown, because in Power BI the relationships of these tables are configured in such a way that the filtering applied in one table is also applied on the other.

**Power Apps** In Power Apps, programming is done through functions very similar to Excel, and through these functions it was possible to apply the same logic mentioned above to find out whether an employee should be visible to the connected manager or not, but instead of filtering the table directly and returning the result table, in the application, I decided to save the filtering results on a list that can be accessed from any screen in the application.

This list was given the name ”employeeList” which will contain all employees that the connected manager has access to and their information contained in the Employee_Details table. In addition to this list there is also the ”chargeCodes” list which contains all active charge codes that are shown on some drop-downs of the application. These lists can be updated manually using the button on the main screen of the application.

### 4.1.3 Database

Initially Excel was used as a database, although Excel is not considered as a database at all, in this project it was used because both Power BI and Power Apps need a premium account to be able to use SQL Server as a database, which only happened in the middle of the work, so in order not to block the development it was decided to use Excel while there was no permission to use a premium account.

After having a premium account I created an instance of the SQL Server locally and the data model that was used in this work, which is represented in the following figure. This data model has a total of 11 tables, which represent not only the existing data on the platforms that will be integrated in the future, but also the tables created by me in order to save the system information.
Figure 4.2: Database schema

All tables will be described next, starting with the tables created by me (Green tables) then it will be the HRM platform table (Red table), followed by the Orange tables (Orange tables) and finally the Salesforce tables (Blue tables).

The tables of the existing platforms were based on what currently exists, in order for the future integration to be smooth, except for the Employee Details table, this table was created just taking into account which fields would be needed for the system.

The existing HRM tables were not taken into account because at the beginning of the work of this thesis it was already known that the company’s HRM platform would be changed, so it was decided that for the development of the thesis it would only be necessary to create a table with all the necessary fields and when the time for integration comes, the data model will be adapted accordingly to the tables that will exist.

New tables

I created 5 new tables where the information contained in them will be about accommodations, flight ticket requirements, charge code associations to employees, notes about
employees and even a table where there will be information about the hierarchy of each employee in the company.

**Accommodation**

Table where requests for accommodation to the Office department are stored.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accommodation_id</td>
<td>float</td>
<td>Primary key</td>
<td>Unique id for each accommodation</td>
</tr>
<tr>
<td>emp_id</td>
<td>integer</td>
<td>Foreign key (Employee_Details)</td>
<td>Unique id for each employee</td>
</tr>
<tr>
<td>charge_code</td>
<td>varchar(100)</td>
<td>Foreign key (ohrm_project)</td>
<td>Unique charge code</td>
</tr>
<tr>
<td>location</td>
<td>varchar(50)</td>
<td>-</td>
<td>Location of the accommodation</td>
</tr>
<tr>
<td>start_date</td>
<td>date</td>
<td>-</td>
<td>Start date of the accommodation</td>
</tr>
<tr>
<td>end_date</td>
<td>date</td>
<td>-</td>
<td>End date of the accommodation</td>
</tr>
<tr>
<td>purchase_order_id</td>
<td>float</td>
<td>-</td>
<td>PO id associated with the accommodation</td>
</tr>
<tr>
<td>update_link</td>
<td>varchar(150)</td>
<td>-</td>
<td>Generated link to insert the PO id</td>
</tr>
<tr>
<td>notes</td>
<td>varchar(max)</td>
<td>-</td>
<td>Notes regarding the accommodation</td>
</tr>
</tbody>
</table>

Table 4.2: Accommodation table

**Flight**

Table where requests for flight tickets to the Office department are stored.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>flight_id</td>
<td>float</td>
<td>Primary key</td>
<td>Unique id for each flight</td>
</tr>
<tr>
<td>emp_id</td>
<td>integer</td>
<td>Foreign key (Employee_Details)</td>
<td>Unique id for each employee</td>
</tr>
<tr>
<td>charge_code</td>
<td>varchar(100)</td>
<td>Foreign key (ohrm_project)</td>
<td>Unique charge code</td>
</tr>
<tr>
<td>origin</td>
<td>varchar(50)</td>
<td>-</td>
<td>Origin city of the flight</td>
</tr>
<tr>
<td>destination</td>
<td>varchar(50)</td>
<td>-</td>
<td>Destination city of the flight</td>
</tr>
<tr>
<td>flight_date</td>
<td>datetime</td>
<td>-</td>
<td>Date and minimum time of the flight</td>
</tr>
<tr>
<td>purchase_order_id</td>
<td>float</td>
<td>-</td>
<td>PO id associated with the flight</td>
</tr>
<tr>
<td>update_link</td>
<td>varchar(150)</td>
<td>-</td>
<td>Link that redirects to the app screen ”Office Input”</td>
</tr>
<tr>
<td>notes</td>
<td>varchar(max)</td>
<td>-</td>
<td>Notes regarding the flight</td>
</tr>
</tbody>
</table>

Table 4.3: Flight table
**Employee_Charge_Code**

Table where the information stored is about which project an employee is on, between what dates, its allocation, work model and whether it needs accommodation or not.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>float</td>
<td>Primary key</td>
<td>Unique id for each employee-charge code association</td>
</tr>
<tr>
<td>emp_id</td>
<td>integer</td>
<td>Foreign key (Employee_Details)</td>
<td>Unique id for each employee</td>
</tr>
<tr>
<td>charge_code</td>
<td>varchar(100)</td>
<td>Foreign key (ohrm_project)</td>
<td>Unique charge code</td>
</tr>
<tr>
<td>work_model</td>
<td>varchar(20)</td>
<td>-</td>
<td>Work model of the employee, can be Nearshore or On site</td>
</tr>
<tr>
<td>allocation</td>
<td>integer</td>
<td>-</td>
<td>Allocation of the employee on the project</td>
</tr>
<tr>
<td>start_date</td>
<td>date</td>
<td>-</td>
<td>Start date of the employee on the project</td>
</tr>
<tr>
<td>end_date</td>
<td>date</td>
<td>-</td>
<td>End date of the employee on the project</td>
</tr>
<tr>
<td>needs_accommodation</td>
<td>binary</td>
<td>-</td>
<td>Flag true or false depending if the employee needs accommodation or not</td>
</tr>
</tbody>
</table>

Table 4.4: Employee_Charge_Code table

**Employee_Notes**

Table where notes that the managers write about the employee are stored.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>note_id</td>
<td>float</td>
<td>Primary key</td>
<td>Unique id for each employee note</td>
</tr>
<tr>
<td>emp_id</td>
<td>integer</td>
<td>Foreign key (Employee_Details)</td>
<td>Unique id for each employee</td>
</tr>
<tr>
<td>notes</td>
<td>varchar(max)</td>
<td>-</td>
<td>Notes regarding the employee</td>
</tr>
</tbody>
</table>

Table 4.5: Employee_Notes table

**User_Access**

This table will be updated by Power BI automatically at each refresh made, through this table it is possible to know the employees each manager has access to and this way restrict the data that each manager can see, both in the dashboards and in the application.
### Table 4.6: User_Access table

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>emp_id</td>
<td>integer</td>
<td>Foreign key (Employee_Details)</td>
<td>Unique id for each employee</td>
</tr>
<tr>
<td>path</td>
<td>nvarchar(255)</td>
<td>-</td>
<td>String generated by the Power BI PATH() function which returns the hierarchy of each employee in the company, this function requires 2 id’s, the id of the employee and the id of the employee’s manager, which can be found on the Employee_Details table.</td>
</tr>
<tr>
<td>level</td>
<td>integer</td>
<td>-</td>
<td>Level of each employee in the company hierarchy that is returned by the Power BI PATHLEVEL() function, the lower the number the higher the level, it requires the path column</td>
</tr>
</tbody>
</table>

**HRM table**

In the development of this project, this table was created to simulate the data existing on the company’s new HRM platform. The data structure of the new platform was not taken into consideration, as it was only introduced in the company in the final part of the work, so it was decided that I would create a table with only the columns needed.

When the integration of my platform with the new HRM platform is done, there will be two possibilities for this to be done, these will be described in the chapter on future work.

**Employee_Details**

Table with all the data related to the employee itself such as name, role, SBU, among others, which are described in the following table:
Chapter 4. Work

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>emp_id</td>
<td>integer</td>
<td>Primary key</td>
<td>Unique id for each employee</td>
</tr>
<tr>
<td>employee_email</td>
<td>varchar(50)</td>
<td>-</td>
<td>Email of the employee</td>
</tr>
<tr>
<td>name</td>
<td>varchar(100)</td>
<td>-</td>
<td>Name of the employee</td>
</tr>
<tr>
<td>role</td>
<td>varchar(30)</td>
<td>-</td>
<td>Role of the employee</td>
</tr>
<tr>
<td>travel_limitation</td>
<td>varchar(5)</td>
<td>-</td>
<td>Yes or No depending if the employee has travel limitations or not</td>
</tr>
<tr>
<td>summary</td>
<td>varchar(max)</td>
<td>-</td>
<td>Summary of the employee</td>
</tr>
<tr>
<td>start_date</td>
<td>date</td>
<td>-</td>
<td>Start date at InnoWave of the employee</td>
</tr>
<tr>
<td>sbu</td>
<td>varchar(20)</td>
<td>-</td>
<td>SBU of the employee</td>
</tr>
<tr>
<td>manager_id</td>
<td>integer</td>
<td>-</td>
<td>Manager id of the employee</td>
</tr>
<tr>
<td>photo_url</td>
<td>varchar(300)</td>
<td>Primary key</td>
<td>Photo url of the employee</td>
</tr>
</tbody>
</table>

Table 4.7: Employee_Details table

Orange tables

On the Orange platform there are two aspects, as explained in the Related Work chapter, one related to the employee’s skills and another part related to charge codes and timesheets. The tables currently in Orange have more columns than those represented here, this does not present any problem in the integration part as the names and data format that are presented in these tables are the same ones I used to carry out my work.

Skill

Table where the skills of each employee are stored.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>skill_id</td>
<td>float</td>
<td>Primary key</td>
<td>Unique id for each employee skill</td>
</tr>
<tr>
<td>emp_id</td>
<td>integer</td>
<td>Foreign key (Employee_Details)</td>
<td>Unique id for each employee</td>
</tr>
<tr>
<td>skill</td>
<td>varchar(50)</td>
<td>-</td>
<td>Name of the skill</td>
</tr>
<tr>
<td>area</td>
<td>varchar(50)</td>
<td>-</td>
<td>Area of the skill</td>
</tr>
<tr>
<td>level</td>
<td>varchar(50)</td>
<td>-</td>
<td>Level of the skill</td>
</tr>
</tbody>
</table>

Table 4.8: Skill table

Language

Table where the languages of each employee are stored.
### Chapter 4. Work

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>language_id</td>
<td>float</td>
<td>Primary key</td>
<td>Unique id for each employee language</td>
</tr>
<tr>
<td>emp_id</td>
<td>integer</td>
<td>Foreign key</td>
<td>Unique id for each employee</td>
</tr>
<tr>
<td>language</td>
<td>varchar(50)</td>
<td>-</td>
<td>Name of the language</td>
</tr>
<tr>
<td>competency</td>
<td>varchar(50)</td>
<td>-</td>
<td>Competency of the language</td>
</tr>
</tbody>
</table>

Table 4.9: Language table

**ohrm_project**

Table where the Orange charge codes are stored.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>varchar(50)</td>
<td>Primary key</td>
<td>Unique charge code name</td>
</tr>
<tr>
<td>end_date</td>
<td>date</td>
<td>-</td>
<td>End date of the charge code on Orange HRM side</td>
</tr>
</tbody>
</table>

Table 4.10: ohrm_project table

**fmr**

Table where the timesheets of each employee are stored.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>float</td>
<td>Primary key</td>
<td>Unique id for each day submitted on the timesheet</td>
</tr>
<tr>
<td>Employee</td>
<td>varchar(50)</td>
<td>Foreign key</td>
<td>Employee name written in the same way as the initial part of the company email</td>
</tr>
<tr>
<td>Project_Name</td>
<td>varchar(50)</td>
<td>Foreign key</td>
<td>Charge code submitted on the timesheet</td>
</tr>
<tr>
<td>Day</td>
<td>date</td>
<td>-</td>
<td>Day of the reported time</td>
</tr>
<tr>
<td>Time_Reported</td>
<td>integer</td>
<td>-</td>
<td>Number of hours worked on the day selected</td>
</tr>
<tr>
<td>Status</td>
<td>varchar(20)</td>
<td>-</td>
<td>Status of the entry, if it is already approved or not</td>
</tr>
</tbody>
</table>

Table 4.11: fmr table

**Salesforce tables**

The tables coming from Salesforce have the same names as the tables I used, but in my version I only have the columns I need, as in the previous case. These tables do not exist in the company’s database but only on Salesforce, so in the integration phase there will be two ways to do it which will be described in the chapter on future works.
Opportunity

Table where the created Salesforce charge codes are stored.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charge_Code_c</td>
<td>varchar(50)</td>
<td>Primary key</td>
<td>Unique charge code</td>
</tr>
<tr>
<td>AccountId</td>
<td>varchar(50)</td>
<td>Foreign key (Account_Name)</td>
<td>Unique account id for each account</td>
</tr>
<tr>
<td>Start_Date_c</td>
<td>date</td>
<td></td>
<td>Start date of the charge code on Salesforce side</td>
</tr>
<tr>
<td>End_Date_c</td>
<td>date</td>
<td></td>
<td>End date of the charge code on Salesforce side</td>
</tr>
</tbody>
</table>

Table 4.12: Opportunity table

Account_Name

Table where all InnoWave’s clients names are stored.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccountId</td>
<td>varchar(50)</td>
<td>Primary key</td>
<td>Unique account id for each account</td>
</tr>
<tr>
<td>Name</td>
<td>varchar(50)</td>
<td>-</td>
<td>Name of the account</td>
</tr>
</tbody>
</table>

Table 4.13: Account_Name table

4.2 Management of SBU resources

This is the first and most important module of this thesis, where the objective is for a manager to easily obtain an overview of their employees and details, as well as being able to change some of this information on the system and finally send or receive emails automatically, improving the manager’s work.

It consists of 3 parts: 4 dashboards in Power BI, a mobile/web application with 4 screens in Power Apps, 9 emails sent automatically to the Office department or to an employee from the application depending on the manager’s actions and 3 emails sent automatically to the managers depending on the existing data in the database, these last two points were accomplished using Power Automate.

4.2.1 Dashboards

In this section I will describe the dashboards created in this module, what information is shown, how it can be filtered and also describe the logic associated with each dashboard. All the different possible layouts of the dashboards are at the appendix A.
**Employees Allocation Overview**

In this dashboard, it is possible for a manager to know the allocation of all his employees in projects. Information about each employee is presented such as their name, manager or role in the company, which can be filtered through the filters present in the dashboard, they are about the information itself and about the time interval shown.

**Available Information**

For each employee, the following information is shown:

- **Details** - Link that redirects the manager to the details screen of an employee on the application.

- **Name** - Employee name.

- **Project End Date** - Project with the most distant end date.

- **Employee Manager** - Employee manager name.

- **SBU** - SBU employee belongs to.

- **Role** - Employee Role.

The allocation of each employee can be seen in the colored squares of the matrix, where each colored square, when the manager passes the mouse over it, can obtain the following information through the Power BI tooltip, in case the square is white, it means that there is no information regarding that day.

- **Green** - In project Total allocation: \(\{1 - 100\}\).
• Yellow - Close to be without a project Total allocation: \(\{1 - 100\}\).

• Red - Without project Total allocation: 0.

Associated with the colors there are several levels of transparency which represent the percentage of the employee’s allocation to the project, which is also divided into 3 groups.

• 0% Transparency - Employee allocated 100% in a project (Green and Yellow) or without any project (Red).

• 20% Transparency - Employee allocated between 50% and 99% in a project (Green and Yellow).

• 50% Transparency - Employee allocated between 1% and 49% in a project (Green and Yellow).

Filtering Options On this dashboard there are 3 types of filters:

1. Date filters - Appear when the manager clicks on the calendar icon on the left side of the matrix.
   
   (a) Week - Weeks to show.
   
   (b) Month - Months to show.
   
   (c) Year - Years to show.

2. Information filters - Appear when the manager clicks on the icon below the calendar on the left side of the matrix.

   (a) Project end date - End date of employee’s projects.

   (b) SBU - Employees SBU.

   (c) Role - Employees Role.

   (d) Allocation - Allocation of employee’s projects.

3. Matrix filters - Visible in the upper right corner of the matrix.

   (a) Columns - The manager can drill down or drill up to change the columns shown.

   (b) Rows - The manager can drill down or drill up to change the rows shown.
Logic Behind  Phrases can be found in each square of the matrix, they are individually calculated using a measure that has been given the name [MatrixOverviewValues].

This measure takes into account the day shown in the matrix column and the employee name in the matrix line, depending on these values, the measure validates if on that day the employee was allocated to a project and if so what was his allocation, if he is allocated to more than one project, the sum of all allocations is made, on the other hand, if he is not allocated to any project, the allocation is 0.

What this measure returns is a phrase that can have the values that were already mentioned above: In project Total allocation: \{1 - 100\}, Close to be without a project Total allocation: \{1 - 100\} or Without project Total allocation: 0, but with the aid of another measure, instead of showing this phrase in each square of the matrix, it takes into account the calculated value of each square and returns the hash-code of the colors: Green, Yellow or Red with the degree of transparency associated, depending on the employee’s allocation on a project.

With this second measure it is possible to change the background color of each square as well as the text color, thus obtaining the final look presented in Employees Allocation Overview.

Employees Per Project

In this dashboard, it is possible for a manager to know the history of all his employees allocation in projects. Information about each employee is presented such as his name, the projects he was allocated on such as the name of the client, which can be filtered using the filters present in the dashboard that are about the information itself.
Available Information  For each employee, the following information is shown in the matrix:

1. Rows
   
   (a) Details - Link that redirects the manager to the details screen of an employee on the application.
   (b) Name - Employee name.
   (c) SBU - SBU employee belongs to.
   (d) Client - Client name associated with the charge code used.
   (e) Project - Charge code used by the employee.
   (f) Allocation - Allocation of the employee on the project.

2. Columns
   
   (a) Employee Start Date - Start date of the employee on the project.
   (b) Employee End Date - End date of the employee on the project.
   (c) Work Model - Employee’s work model on the project.

Filtering Options  In this dashboard there are 2 types of filters:

1. Information filters - Appear when the manager clicks on the icon on the left side of the matrix.
   
   (a) Client Name - Names of each project’s clients.
   (b) Manager - Employee Manager.
   (c) Role - Employee Role.

2. Matrix filters - Visible in the upper right corner of the matrix.
   
   (a) Rows - The manager can drill down or drill up to change the rows shown.

Logic Behind  The information presented in the matrix comes from 3 tables: Employee_Charge_Code, Employee_Details and Account_Name, through the capabilities of Power BI it was only necessary to drag the columns I wanted to show on the matrix, taking into account if this information will be in the rows or if they will be in the columns, since the tables are relational it is possible to use columns from several tables in the same matrix.
Employees Abroad

In this dashboard, it is possible for a manager to know which of his employees are working abroad on projects. The information presented is the employee’s name and SBU, as well as when he has an accommodation booked and scheduled airplane flight, which can be changed through the filters present in the dashboard that are about the time range shown.

![Employees Abroad](image)

**Figure 4.5: Employees Abroad**

**Available Information**  For each employee abroad, the following information is shown:

- **Details** - Link that redirects the manager to the abroad details screen of an employee on the application.
- **Name** - Employee name.
- **SBU** - SBU employee belongs to.

Information abroad of each employee can be seen in the colored squares of the matrix, where each colored square, when the manager passes the mouse over it, can obtain the following information through the Power BI tooltip, in case the square is white it means that there is no information regarding that day.

- **Green** - Accommodation On:{Accommodation city} Charge code: {Associated charge code}.
- **Yellow** - Accommodation close to end date On:{Accommodation city} Charge code: {Associated charge code}.
- **Blue** - Flight From:{Departure city} To:{Arrival city} Charge code: {Associated charge code}.
• Red - Flight needed for accommodation On: {Accommodation city} Charge code: {Associated charge code}.

Filtering Options In this dashboard there are 2 types of filters:

1. Date filters - Appear when the manager clicks on the calendar icon on the left side of the matrix.
   (a) Week - Weeks to show.
   (b) Month - Months to show.
   (c) Year - Years to show.

2. Matrix filters - Visible in the upper right corner of the matrix.
   (a) Columns - The manager can drill down or drill up to change the columns shown.
   (b) Rows - The manager can drill down or drill up to change the rows shown.

Logic Behind What is found in each square of the matrix are phrases that are individually calculated using a measure named [MatrixAbroadValues].

This measure takes into account the day in the matrix column and the employee in the matrix line, depending on these values, the measure validates if on that day the employee has a scheduled flight, if there is a reservation of any accommodation, if it is close to the end date of the contract and finally validates if it is necessary to make a request for a flight ticket to the Office department.

What this measure returns is a phrase that can have the values already mentioned above: Accommodation On: {Accommodation city} Charge code: {Associated charge code}, Accommodation close to end date On: {Accommodation city} Charge code: {Associated charge code}, Flight From: {Departure city} To: {Arrival city} Charge code: {Associated charge code} or Flight needed for accommodation On: {Accommodation city} Charge code: {Associated charge code}, but with the help of another measure, instead of showing this phrase in each square of the matrix, it takes into account the calculated value of each square and returns the hash-code Colors: Green, Yellow, Blue or Red.

With this second measure it is possible to change the background color of each square as well as the text color, thus obtaining the final look presented in Employees Abroad.

Employees Overview KPIs

This dashboard presents the manager with a graphic and multiple important values representing the KPIs of an SBU. The information presented are: how many employees were
without a project at a given time or information about employees abroad, which can be changed through the filters present in the dashboard that are about the time interval shown.

Figure 4.6: Employees Overview KPIs

Available Information  The graphic present in this dashboard represent the following information based on the time interval selected:

1. Without a project - Number of not allocated employees on a project.
2. Partially allocated - Number of employees partially allocated on a project.
3. Fully allocated - Number of employees fully allocated on a project.
4. Employees - Total number of employees.
5. Employees abroad - Total number of employees abroad.

The values also displayed in this dashboard are:

1. Close to be without a project - Number of employees close to being without a project.
2. Accommodation close to end date - Number of employees whose accommodation is close to the end date and there is no flight ticket booked.
3. Flight tickets needed for the next week - Number of employees whose accommodation is close to the start date and there is no reserved flight ticket.
**Filtering Options**  In this dashboard there is only 1 type of filter:

1. Date filters - Appear when the manager clicks on the calendar icon on the left side of the matrix.

   (a) Week - Weeks to show.

   (b) Month - Months to show.

   (c) Year - Years to show.

**Logic Behind**  For each information shown in this line graph there is a measure that calculates the respective number of employees taking into account the dates displayed, for the 3 values the current date is taken into consideration instead.

### 4.2.2 Application

This section describes the entire application developed, which will contain the information, features and logic implemented on each screen. All the different possible layouts of the application are at the appendix B.

#### Employees

This is the main screen of the application and the initial screen that the manager will see when opening the application, here a list is shown with the name, role, SBU and work model of the current project, if any, of each employee as well as the photo. This list is dependent on the manager and their access level. The manager is provided with several filters that can be used in order to find the employee he wants easily, after finding him, he can simply click on the rectangular area where the employee’s information is and he will be redirected to the Employee Details screen.
Available Information  On this screen, the manager has a list of all employees on whom he can make project changes or make requests to the Office department. The following information is shown:

- Employee Photo - Employee photo.
- Name - Employee name.
- Role - Employee Role.
- SBU - SBU employee belongs to.
- Work Model - Work Model of the employee’s current project if allocated to one, otherwise it is shown Without a project.

Features  The manager on this screen can easily filter the list of employees using the following filters:

1. Work Model Drop Down - Depending on the selected value of the work model the displayed employees change.
   
   (a) All work models - Nearshore and On site models are taken into account in the employees list.
   
   (b) Nearshore - Only employees currently working on a nearshore model.
   
   (c) On site - Only employees currently working in on site model.
   
   (d) Without a project - Employees currently without a project.
2. **SBU Drop Down** - Depending on the selected SBU value the displayed employees change.

   (a) All SBUs - Employees from all SBUs.

   (b) Enterprise Solutions - Only SBU Enterprise Solutions employees.

   (c) Iot - Only SBU Iot employees.

   (d) Quality Assurance - Only SBU Quality Assurance employees.

   (e) TV & Media - Only SBU TV employees & Media.

3. **Search Box** - Depending on the text entered, the manager can search by the name or role of the employee.

   In addition to these types of filtering, the manager is also provided with a refresh button which updates the list of employees when pressed.

**Logic Behind**  The list of employees shown on this screen depends on the connected manager, for an employee to appear in the list he needs to be in a hierarchical level below the manager and belong to the same SBU as him or be allocated to a project in the manager’s SBU, only if these validations are true the employee is shown otherwise it is filtered out, in this way the manager can only make changes on the employees he has access to.

   When clicking on the icon in the upper right corner, two lists are updated. The first list is the list of employees to which the manager has access taking into account the rules described above, the second list is the list of charge codes that can be associated with an employee in the following screens, this one only has one rule, for one charge code to be valid it must have its end date in the future, it is possible for all managers to have access to all charge codes, even if they belong to a SBU other than the manager’s, this will allow an employee to be allocated to a project from another SBU if necessary.

**Employee Details**

On this screen, the manager can see information related to the employee as well as edit some of it. On this screen the manager can allocate a employee to a new project, edit the allocation of a current or future project and can also remove the association of a project. In addition to these features, the manager can navigate to the screen where requests are made to the Office department.
Chapter 4. Work

Figure 4.8: Employee Details

**Available Information**  On this screen, the manager has various information about the employee such as his name or which projects he has already been on. Some of this information can be changed by the manager, such as the notes about the employee. Finally, it is possible to navigate to the screen where there is information about requests made to the Office department. The following information is shown:

1. Employee’s photo - Photo of the selected employee.
2. Name - Name of the selected employee.
3. Role - Role of the selected employee.
4. Summary - Summary of the selected employee.
5. Notes - Notes about the selected employee.
6. Manager - Manager of the selected employee.
7. Travel limitations - Travel limitations of the selected employee.
8. InnoWave start date - InnoWave’s start date of the selected employee.
9. Projects table - Table with the various projects the employee has been, is or will be on.
   - (a) Status - Project status that can have one of the following values: Old, Current or Future.
   - (b) Project - Charge code associated with the employee.
(c) Work Model - Project work model associated with the employee.
(d) Allocation - Allocation of the employee on the project.
(e) Start Date - Project start date associated with the employee.
(f) End Date - End date of the project associated with the employee.
(g) Edit - Allows the manager to edit the association of a project, only possible in projects with Current or Future status.
(h) Remove - Allows the manager to remove the association of a project, it is only possible in projects with Future status.

10. Skills table - Table with all the employee’s skills.
   (a) Area - Area of the skill.
   (b) Skill - Skill name.
   (c) Level - Skill knowledge level.

11. Language table - Table with all languages that the employee knows.
   (a) Language - Language name.
   (b) Competency - Language Competency.

Features  The manager on this screen can edit information related to the selected employee and navigate to another screen or to Microsoft Teams, these features are described below:

1. Notes - When the manager clicks on the icon represented by a pencil in the upper right corner, it activates the functionality to edit the notes that are in the lower left corner of the screen.

2. Allocation in projects - In this screen there are 4 functions that can be done in relation to the allocation of the employee in projects.
   (a) Details of a project - By clicking on any of the table rows and any column, except the Edit and Remove columns, the manager can see the project details.
   (b) New project - Allocate the employee to a new project.
   (c) Edit project - Edit the employee allocation in an existing project.
   (d) Remove project - Remove the employee’s allocation in a project.

3. Teams - Pressing the button ”Send message on teams” on the application, it opens Microsoft Teams in a private chat with the employee.

4. Abroad details - Pressing the ”View abroad details” button the application redirects the manager to the Employee Abroad Details screen.
Logic Behind  The manager can make changes to the employee’s allocation in projects, depending on the action the manager wants to do there are some rules that must be followed, these will be described within each possible flow:

1. Add new project - Status: Future.
   (a) The charge code cannot already be associated with the employee.
   (b) The start and end dates entered by the manager cannot exceed the dates of the charge code itself.
   (c) The start date cannot be earlier than the current day.
   (d) The allocation cannot exceed the value 100, other projects in parallel are taken into account.

2. Edit Project - Status: Current.
   (a) The charge code cannot be changed.
   (b) The start and end dates entered by the manager cannot exceed the dates of the charge code itself.
   (c) If the work model or allocation is changed, the start date represents when the changes to the association with the project will take effect.
   (d) The start date can only be changed if the work model or allocation has been changed.
   (e) The end date can be changed without changing any other information.
   (f) The allocation cannot exceed the value 100, other projects in parallel are taken into account.

   (a) The charge code cannot be changed.
   (b) The start and end dates entered by the manager cannot exceed the dates of the charge code itself.
   (c) The work model can be changed without any other information being changed.
   (d) The start date can be changed without any other information being changed.
   (e) The end date can be changed without changing any other information.
   (f) The allocation cannot exceed the value 100, other projects in parallel are taken into account.

4. Remove project - Status: Future.
Employee Abroad Details

On this screen the manager can view, edit and remove information related to requests made to the Office department for an accommodation or airplane tickets.

![Employee Abroad Details](image)

**Figure 4.9: Employee Abroad Details**

**Available Information** The information shown on this screen is divided into two tables: Accommodation booking and Flight booking respectively, which will be described below:

1. **Accommodation booking** - Table with the various accommodations the employee has been on.
   
   (a) **Status** - Accommodation status which can have one of the following values: Old, Current or Future.
   
   (b) **Charge Code** - Charge code associated with the accommodation.
   
   (c) **Location** - Location of the accommodation.
   
   (d) **Start Date** - Accommodation start date.
   
   (e) **End Date** - Accommodation end date.
   
   (f) **Purchase Order Id** - Id of the PO created for this accommodation.
   
   (g) **Edit** - Allows the manager to edit the accommodation, it is only possible in accommodations with the status Current or Future.
   
   (h) **Cancel** - Allows the manager to cancel the accommodation, it is only possible in accommodations with Future status.

2. **Flight booking** - Table with the various flight tickets that the employee has had.
(a) Status - Flight ticket status that can have one of the following values: Old or Future.

(b) Charge Code - Charge code associated with the flight ticket.

(c) Origin - Origin city for the flight ticket.

(d) Destination - Destination city for the flight ticket.

(e) Flight Date - Minimum date and time for the flight ticket.

(f) Purchase Order Id - Id of the PO created for this flight ticket.

(g) Edit - Allows the manager to edit the flight ticket, it is only possible on flight tickets with Future status.

(h) Cancel - Allows the manager to cancel the flight ticket, it is only possible on flight tickets with Future status.

Features  The manager on this screen can request, edit and cancel requests made to the Office department about accommodation or flight tickets, with the following possible scenarios:

1. Accommodation Orders - There are 4 features relating to accommodations on this screen.

   (a) Details of an accommodation - By clicking on any of the rows of the table and any column, except the columns Edit and Cancel, the manager can see the details of the accommodation.

   (b) Request an accommodation - Request a new accommodation for the employee by clicking on the "Request new accommodation order” button.

   (c) Request changes to an accommodation - Request changes to an existing accommodation by clicking on the Edit text if it is present in the column.

   (d) Cancel an accommodation - Cancel an existing accommodation by clicking on the Cancel reservation text if it is present in the column.

2. Flight Orders - There are 4 flight ticket features on this screen.

   (a) Details of a flight ticket - By clicking on any of the rows of the table and any column, except the columns Edit and Cancel, the manager can see the details of the flight ticket.

   (b) Request a flight ticket - Request a new flight ticket for the employee, clicking on the "Request new flight order” button, it can be a one-way ticket or a return ticket.
(c) Request changes to a flight ticket - Request changes to an existing flight ticket by clicking on the Edit text if it is present in the column.

(d) Cancel a flight ticket - Cancel an existing flight ticket by clicking on the Cancel reservation text if it is present in the column.

**Logic Behind** The manager can make requests about accommodations to the Office, depending on the action the manager wants to do, there are some rules that must be followed, which are described below:

1. Request an accommodation - Status: Future.
   
   (a) Any charge code present in the drop down can be used.
   
   (b) The start and end dates must belong to a valid time range in the future.
   
   (c) Both the city of accommodation and extra information for the Office department are mandatory.

2. Request changes to an accommodation - Status: Current.
   
   (a) The charge code cannot be changed.
   
   (b) The start date cannot be changed.
   
   (c) The end date can be changed only with the rule that it must be at least the day after the current day.
   
   (d) Accommodation city cannot be changed.
   
   (e) Extra information for Office department is mandatory.

3. Request changes to an accommodation - Status: Future.
   
   (a) The charge code cannot be changed.
   
   (b) Start and end dates can be changed but must belong to a valid time range in the future.
   
   (c) Accommodation city cannot be changed.
   
   (d) Extra information for Office department is mandatory.

4. Request the cancellation of an accommodation - Status: Future.
   
   (a) Extra information for Office department is mandatory.

It is also possible to make requests for flight tickets to the Office, depending on the action that the manager wants to do, there are some rules that must be followed, which are described below:
1. Request a flight ticket - Status: Future.

   (a) Any charge code present in the drop down can be used.
   (b) Both the city of origin and destination are required.
   (c) The date of the flight ticket cannot be earlier than the current day.
   (d) Extra information for Office department is mandatory.

2. Request a return flight - Status: Future.

   (a) Any charge code present in the drop down can be used.
   (b) Both the city of origin and destination are required.
   (c) The date of the outward flight ticket cannot be earlier than the current day.
   (d) The date of the return flight cannot be earlier than the date of the outward flight.
   (e) Extra information for Office department is mandatory.

3. Request changes to a flight ticket - Status: Future.

   (a) The charge code cannot be changed.
   (b) The cities of origin and destination cannot be changed.
   (c) The date of the flight ticket cannot be earlier than the current day.
   (d) Extra information for Office department is mandatory.


   (a) Extra information for Office department is mandatory.

**Office Input**

This screen is accessed through the link contained in the emails automatically sent to the Office about an accommodation or flight tickets. Here it is possible for the Office to update the request made with the Id of the PO created for it, change which is reflected in the Employee Abroad Details screen.
Available Information  The information shown on this screen is about accommodation orders or flight orders, which will be described below:

1. Accommodation order update - Order about an accommodation
   
   (a) Charge Code - Label with the charge code associated with the accommodation.
   
   (b) Location - Label with the location of the accommodation.
   
   (c) Start Date - Label with the accommodation start date.
   
   (d) End Date - Label with the accommodation end date.
   
   (e) Purchase Order Id - Text field where Office will enter the PO Id.

2. Flight order update - Order about a flight ticket, which depending on the request made to the Office, may also have the return flight.

   (a) Charge Code - Label with the charge code associated with the flight ticket.
   
   (b) Origin - Label with the origin city of the flight ticket.
   
   (c) Destination - Label with the destination city of the flight ticket.
   
   (d) Flight Date - Label with the minimum date and time of the flight ticket.
   
   (e) Purchase Order Id - Text field where the Office will enter the PO Id.

Features   The Office on this screen can add the PO Id to the request made by a manager as soon as it is created, the orders can either be about an accommodation or a flight ticket.

Logic Behind   When the ”Update” button is pressed, the PO ID is stored in the database.
### 4.2.3 Automated Flows

In this section all automatic email flows of this module sent through Power Automate will be described, I will describe what information is sent in each one and in what situations they are sent.

#### New charge code

This email will be sent to the employee to whom the manager has associated a new charge code, it is only sent if the manager explicitly indicates it in the application.

**Figure 4.11: New charge code**

**New charge code (2021PTALTPT08TM - Shared IoT)**

![Profile picture]

Pedro Nuno Martins  
To: Pedro Nuno Martins

Hello,

Between the days **01 November 2021** and **31 December 2021** please use the following charge code.

**Charge code: 2021PTALTPT08TM - Shared IoT.**

Thank you.

Available Information  From the title of the email it is possible to understand that this is a new charge code that the employee will use but also the charge code itself is shown, this makes it easier to do a search in the email tool to find all emails related with this charge code.

The following information can be found in the contents of the email:

- Start date - Date from when the employee must start using the charge code.
- End date - Date from when the employee cannot use the charge code.
- Charge code - Charge code to be used in the timesheet.

How it is triggered  The email is triggered on the Employee Details screen when the manager associates a new charge code to the employee and clicks on the check box to notify him.

#### Update to a charge code

This email will be sent to the employee when the manager makes changes to the charge code association, it is only sent if the manager explicitly indicates it in the application.
Chapter 4. Work

Figure 4.12: Update to a charge code

Available Information  From the title of the email it is possible to understand that this is an update to the charge code that will be used by the employee but also the charge code itself is shown, this makes it easy to do a search in the email tool to find all emails related with this charge code.

The following information can be found in the contents of the email:

- Charge code - Charge code to be used in the timesheet.
- Start date - New date from when the employee must start using the charge code.
- End date - New date from when the employee cannot use the charge code.

How it is triggered  The email is triggered on the Employee Details screen when the manager makes some changes on any employee charge code association and clicks on the check box to notify him.

Request new accommodation

This email will be sent to the Office department when the manager wants to make a request for a new accommodation for a employee, the email is always sent when a request is made in the application, it is also possible to add emails in CC to the sent email.
Figure 4.13: Request new accommodation

**Available Information**  From the title of the email it is possible to understand that this is a request for a new accommodation for the employee, the name of the employee and the charge code associated with it are also shown, this makes it easy to do a search in the email tool to find all emails related with this charge code.

The following information can be found in the contents of the email:

- CC - Emails to whom the email is also sent.
- Employee name - Name of the employee.
- Accommodation city - City of the employee’s accommodation.
- Accommodation start date - Start date of the employee’s accommodation.
- Accommodation end date - End date of the employee’s accommodation.
- Accommodation charge code - Charge code to be associated with the accommodation.
- Accommodation details - Details entered into the application by the manager.
- PO ID - Link that redirects the Office to the Office Input screen.

**How it is triggered**  This email is triggered on the Employee Abroad Details screen when the manager requests a new accommodation for an employee.
Chapter 4. Work

Request update to existing accommodation

This email will be sent to the Office department when the manager wants to make a change to an accommodation for an employee, the email is always sent when a request is made in the application, it is also possible to add emails in CC to the sent email.

![Email Sample](image)

Hello,

There were changes to the accommodation for Nuno Reis which is associated with the charge code 2021PTALTTP08BM - Shared IoT at Brussels.

New dates to take into account:
06 September 2021 up to 30 November 2023.

Additional information:
Information provided by the manager on the application

Thank you.

Figure 4.14: Request update to existing accommodation

Available Information  From the title of the email it is possible to understand that this is a request to change the employee’s accommodation, the name of the employee and the charge code associated with it are also shown, this makes it easy to do a search in the email tool to find all emails related with this charge code.

The following information can be found in the contents of the email:

- **CC** - Emails to whom the email is also sent.
- **Employee name** - Name of the employee.
- **Accommodation city** - City of the employee’s accommodation.
- **Accommodation start date** - Start date of the employee’s accommodation.
- **Accommodation end date** - End date of the employee’s accommodation.
- **Accommodation charge code** - Charge code to be associated with the accommodation.
- **Accommodation details** - Details entered into the application by the manager.

How it is triggered  The email is triggered on the Employee Abroad Details screen when the manager requests a change to an employee’s accommodation.
Request cancellation of existing accommodation

This email will be sent to the Office department when the manager wants to cancel an employee’s accommodation, the email is always sent when a request is made in the application, it is also possible to add emails in CC to the sent email.

Available Information

From the title of the email it is possible to understand that this is a request to cancel the employee’s accommodation, the name of the employee and the charge code associated with it are also shown, this makes it easy to do a search in the email tool to find all emails related with this charge code.

The following information can be found in the contents of the email:

- CC - Emails to whom the email is also sent.
- Employee name - Name of the employee.
- Accommodation charge code - Charge code to be associated with the accommodation.
- Accommodation city - City of the employee’s accommodation.
- Accommodation start date - Start date of the employee’s accommodation.
- Accommodation end date - End date of the employee’s accommodation.
- Accommodation details - Details entered into the application by the manager.

How it is triggered

The email is triggered on the Employee Abroad Details screen when the manager makes a request to cancel an employee’s accommodation.
Request one way flight ticket

This email will be sent to the Office department when the manager wants to make a request for a new one-way flight ticket for an employee, the email is always sent when a request is made in the application, it is also possible to add emails in CC to the sent email.

One-way flight for Nuno Reis (2021PTGLPPT02TM - Shared QA)

Hello,

I need quotes for a flight on behalf of Nuno Reis.

One-way flight:
From Lisbon to Bruxells for day 06 September 2021 starting from 09h00.

The cost of the flight must be allocated to the charge code: 2021PTGLPPT02TM - Shared QA.

Details to consider:
Information provided by the manager on the application

Please send me proposals for me to validate.

After creating the PO, please add the PO ID at this link: Insert PO ID

Thank you.

Figure 4.16: Request one way flight ticket

Available Information  From the title of the email it is possible to understand that this is a request for a new one-way flight ticket to the employee, the name of the employee and the charge code associated with it are also shown, this makes it easy to do a search in the email tool to find all emails related with this charge code.

The following information can be found in the contents of the email:

• CC - Emails to whom the email is also sent.

• Employee name - Name of the employee the flight ticket is for.

• Origin city - Origin city of the flight ticket.

• Destination city - Destination city of the flight ticket.

• Flight date - Minimum date and time of the flight ticket.

• Flight charge code - Charge code to be associated with the flight ticket.
• Flight details - Details entered into the application by the manager.
• PO ID - Link that redirects the Office to the Office Input screen.

**How it is triggered**  The email is triggered on the Employee Abroad Details screen when the manager requests a new flight ticket for an employee.

**Request return flight ticket**

This email will be sent to the Office department when the manager wants to place a request for a new return flight for an employee, the email is always sent when a request is made in the application, it is also possible to add emails in CC to the sent email.

**Round-trip flights to Nuno Reis (2021PTALTPT07TM - Shared IoT)**

![Pedro Nuno Martins](image)

To Pedro Nuno Martins

Hello,

I need quotes for two flights on behalf of Nuno Reis.

One-way flight:
From Lisbon to Bruxells for day **06 September 2021** starting from **10h00**.

Return flight:
From Bruxells to Lisbon for day **30 September 2021** starting from **10h00**.

The cost of flights must be allocated to the charge code: **2021PTALTPT07TM - Shared IoT**.

Details to consider:
Information provided by the manager on the application

Please send me proposals for me to validate.

After creating the PO, please add the PO ID at this link: [Insert PO ID]

Thank you.

**Figure 4.17: Request return flight ticket**

**Available Information**  From the title of the email it is possible to understand that this is a return flight ticket request for the employee, the name of the employee and the charge code associated with it are also shown, this makes it easy to do a search in the email tool to find all emails related with this charge code.

The following information can be found in the contents of the email:
• CC - Emails to whom the email is also sent.

• Employee name - Name of the employee the flight ticket is for.

• Origin city outbound flight - Origin city of the outbound flight ticket.

• Destination city outbound flight - Destination city for the outbound flight ticket.

• Flight date outbound flight - Minimum date and time of the outbound flight.

• Origin city return flight - Origin city of the return flight ticket.

• Destination city return flight - Destination city for the return flight ticket.

• Flight date return flight - Minimum flight date and time of return flight.

• Flight charge code - Charge code to be associated with the flight ticket.

• Flight details - Details entered into the application by the manager.

• PO ID - Link that redirects the Office to the Office Input screen.

**How it is triggered**  The email is triggered on the Employee Abroad Details screen when the manager makes a return flight request for an employee.

**Request update to existing flight ticket**

This email will be sent to the Office department when the manager wants to make a change to a flight ticket of an employee, the email is always sent when a request is made in the application, it is also possible to add emails in CC to the sent email.

![Flight changes for Nuno Reis (2021PTGLPPT02TM - Shared QA)](image)

Hello,

There were changes in the flight for Nuno Reis which is associated with the charge code 2021PTGLPPT02TM - Shared QA in which it leaves from Lisbon to Brussels.

New minimum flight time:
07 September 2021 09:00.

Additional information:
Information provided by the manager on the application.

Thank you.

Figure 4.18: Request update to existing flight ticket
Available Information  From the title of the email it is possible to understand that this is a request to change an employee’s flight ticket, the name of the employee and the charge code associated with it are also shown, this makes it easy to do a search in the email tool to find all emails related with this charge code.

The following information can be found in the contents of the email:

- CC - Emails to whom the email is also sent.
- Employee name - Name of the employee the flight ticket is for.
- Flight charge code - Charge code to be associated with the flight ticket.
- Origin city - Origin city of the flight ticket.
- Destination city - Destination city of the flight ticket.
- New flight end date - New minimum date and time of the flight ticket.
- Flight details - Details entered into the application by the manager.

How it is triggered  The email is triggered on the Employee Abroad Details screen when the manager requests a change to an employee’s flight ticket.

Request cancellation of existing flight ticket

This email will be sent to the Office department when the manager wants to cancel an employee’s flight ticket, the email is always sent when a request is made in the application, it is also possible to add emails in CC to the sent email.

Figure 4.19: Request cancellation of existing flight ticket

Available Information  From the title of the email it is possible to understand that this is a request to cancel the employee’s flight ticket, the name of the employee and the charge code associated with it are also shown, this makes it easy to do a search in the email tool to find all emails related with this charge code.
The following information can be found in the contents of the email:

- **CC** - Emails to whom the email is also sent.

- **Employee name** - Name of the employee the flight ticket is for.

- **Flight charge code** - Charge code to be associated with the flight ticket.

- **Origin city** - Origin city of the flight ticket.

- **Destination city** - Destination city of the flight ticket.

- **Flight date** - Minimum date and time of the flight flight.

- **Flight details** - Details entered into the application by the manager.

**How it is triggered**  The email is triggered on the Employee Abroad Details screen when the manager makes a request to cancel an employee’s flight ticket.

**Employee Charge code reaching end date**

This email is sent to the various managers of the SBUs informing which employees are close to the end date of a project and whether or not they have a project in the near future. The multiple emails are automatically sent by Power Automate, depending on the values returned by a query stored in the database.
Chapter 4. Work

Figure 4.20: Employee Charge code reaching end date

Available Information  The title of the email informs the manager that this is an email about employees close to be without a project.

In the content of the email there is a list of employees with the following details:

- Employee name - Employee name close to be without a project.
- Current project - Charge code near the end date defined.
- Allocation - Allocation of the employee on the project.
- Work model - Work model of the employee on the project.
- End date - Charge code end date defined.
• Future project - Depending on whether or not the employee has a project after this one, if there is, the starting date of the next project is indicated.

• Link - Redirects the manager to the Employee Details screen of the selected employee.

**How it is triggered**  
The email is automatically triggered by Power Automate, as this is a scheduled cloud flow that is configured to run once a day at 2 am, a value that can be easily modified.

This flow starts by executing a procedure stored in the database with the name EmployeeProjectEndDateWarning, which makes several statements and in the end returns the list of employees close to be without a project as well as the necessary information to be presented in the email. After obtaining the list, this is divided by manager and the information of the employees of each manager is transformed to what can be seen in ?? and finally the emails are sent to the various managers.

**Accommodation reaching end date**

This email is sent to the various managers of the SBUs informing them which accommodations are close to their end date. The multiple emails are automatically sent by Power Automate, depending on the values returned by a query stored in the database.

![Accommodations reaching the end date](image)

**Available Information**  
The title of the email informs the manager that this is an email about accommodations close to their end date.

In the content of the email there is a list of employees with the following details:

• Employee name - Name of the employee with accommodation close to its end date.

• Accommodation end date - Accommodation end date.
• Location - City of the employee’s accommodation.

• Accommodation charge code - Charge code of the employee’s accommodation.

• Link - Redirects the manager to the Employee Abroad Details screen of the selected employee.

**How it is triggered**  The email is automatically triggered by Power Automate, as this is a scheduled cloud flow that is configured to run once a day at 2 am, a value that can be easily modified.

This flow starts by executing a procedure saved in the database with the name AccommodationWarning, which makes several statements and at the end returns the list of employees with accommodation close to the end date, as well as the necessary information to be presented in the email. After obtaining the list, this is divided by manager and the information of the employees of each manager is transformed to what can be seen in **Accommodation reaching end date** and finally the emails are sent to the various managers.

**Flight ticket needed to be booked**

This email is sent to the various managers of the SBUs informing them which flight tickets are necessary to request. The various emails are automatically sent by Power Automate, depending on the values returned by a query stored in the database.

![Figure 4.22: Flight ticket needed to be booked](image)

**Available Information**  The title of the email informs the manager that this is an email about missing flight tickets.

In the content of the email there is a list of employees with the following details:

• Employee name - Name of the employee who needs a flight ticket.

• Accommodation start date - Accommodation start date which should be the date of the flight ticket.
• Location - City of the employee’s accommodation.

• Accommodation charge code - Charge code associated with the employee’s accommodation.

• Link - Redirects the manager to the Employee Abroad Details screen of the selected employee.

**How it is triggered**  The email is automatically triggered by Power Automate, as this is a scheduled cloud flow that is configured to run once a day at 2 am, a value that can be easily modified.

This flow starts by executing a procedure saved in the database with the name MissingFlightWarning, which makes several statements and in the end returns the list of employees with missing flight tickets, as well as the necessary information to present in the email. After obtaining the list, this is divided by manager and the information of the employees of each manager is transformed to what can be seen in [Flight ticket needed to be booked](#) and finally the emails are sent to the various managers.

### 4.3 Projects Overview

On the second module of this thesis, the objective is that it should possible for a manager to easily obtain a global view of their projects, details and receive emails automatically, improving their work. This module consists of 2 parts: 1 dashboard in Power BI and 1 email sent automatically to managers depending on the data in the database which uses Power Automate. All the different possible layouts of the dashboard are at the appendix A.

#### 4.3.1 Dashboards

**Projects Overview**

In this dashboard, it is possible for a manager to know the history of all the projects that his employees have worked on, information about each project is presented, such as the charge code, the end date and the names of the employees who used the charge code, the information displayed can be changed through the filters present in the dashboard that are about the data itself and about the time interval shown.
Available Information  For each project, the following information is shown in the matrix:

1. Rows
   (a) SBU - SBU to which the project belongs.
   (b) Client Name - Name of the client of the project.
   (c) Project End Date - End date of the charge code.
   (d) Project - Charge code.
   (e) Employee Name - Employees using the charge code.

2. Columns
   (a) Employees - Total number of employees using the charge code.

In addition to the matrix, there is also a graph in which the following values are displayed:

- Clients - Total number of clients over time.
- Projects - Total number of projects over time.
- Employees In Projects - Total number of employees in projects over time.
Filtering Options  On this dashboard there are 3 types of filters:

1. Date filters - Appears when the manager clicks on the calendar icon on the left side of the matrix.
   
   (a) Week - Weeks to show.
   
   (b) Month - Months to show.
   
   (c) Year - Years to show.

2. Information filters - Appears when the manager clicks on the icon below the calendar on the left side of the matrix.
   
   (a) Client Name - Name of the client of the project.
   
   (b) Project end date - End date of the charge code.

3. Matrix filters - Visible in the upper right corner of the matrix.
   
   (a) Rows - The manager can drill down or drill up to change the rows shown.

Logic Behind  The data presented in the matrix comes from 4 tables: ohrm_project, Opportunity, Employee_Details and Account_Name, through the capabilities of Power BI it was only necessary to drag the columns I wanted to show into the matrix, taking into account if this information will be in the rows or if they will be columns, this is only possible since the tables are relational.

For each graph line shown in this dashboard, there is a measure that calculates the respective number of clients, projects and employees in projects, taking into account the day shown in the chart.

4.3.2  Automated Flows

In this section the email flow sent automatically through Power Automate is described, I will describe what information is sent and in what situations it is sent.

**Charge code reaching end date**

This email is sent to the various managers of the SBUs informing them that charge codes are close to their end date. The various emails are automatically sent by Power Automate, depending on the values returned by a query stored in the database.
Figure 4.24: Charge code reaching end date

**Available Information**  The title of the email informs the manager that this is an email about charge codes close to its end date.

In the content of the email there is a list of charge codes with the following details:

- Charge code - Charge code near its end date.
- End date - End date of the charge code.

**How it is triggered**  The email is automatically triggered by Power Automate, as this is a scheduled cloud flow that is configured to run once a day at 2 am, a value that can be easily modified.

This flow starts by executing a procedure stored in the database with the name ProjectEndDateWarning, which makes several statements and at the end returns the charge code list close to the end date as well as the necessary information to present in the email. After obtaining the list, this is divided by manager, the information of the employees of each manager is transformed to what can be seen in the previous figure and finally the emails are sent to the various managers.

### 4.4 Automated processes

This section describes how the automatic data update in the Employee_Charge_Codes table is done, as well as a possible solution for the automatic pre-fill of each employee’s timesheets.
4.4.1 Employees projects

In order for my system to be as autonomous as possible I decided to make a small program in Java which reads information from my database tables of Orange and WeGrow and it updates the Employee_Charge_Codes table.

This program is run from the command line and depending on the arguments passed it has two possible flows.

- The first flow aims to populate the new table with all the history of charge codes used by each employee since joining InnoWave, this way once the system goes live it is possible to immediately use it and obtain information about the current status of each SBU.

- The second flow is practically the same as the previous one but with the difference that instead of the program looking at the entire charge code history of each employee, in this flow it will only look at the last 7 days of timesheets introduced by the employees and just insert in the new table if the association does not already exist.

Through these two flows it is possible for my system to keep up to date even without the interaction by the managers in the application, with the disadvantage that the end date or allocation may not correspond exactly to reality, but this informations can be edited by the manager easily and quickly in the application.

On the next steps I describe all the logic behind the program:

1. Fetch all employees from Employee_Details tables and iterate over all of them.

   (a) Fetch all hours introduced by the current employee on the timesheets from the fmr table depending on the flow.

   (b) Fetch all existing project associations of the current employee from the Employee_Charge_Code table.

   (c) Filter hours logged based on the status equals APPROVED and charge code not existing on a black list of charge codes.

   (d) Fetch all end dates from the ohrm_projects table based on the charge codes existing on the hours logged.

   (e) Convert the hours logged into objects of the same type as the Employee_Charge_Code table entries and iterate over this list.

   i. If there are no intersections between the current project and either a new project from hours logged or an already existing association on my table, a new entry is added of the current project with 100 allocation.
ii. If there are intersections, a calculation of the ranges between the intersected projects is done. This will result in multiple entries depending on the number of concurrent projects and when they intersect, this is done in order to obtain the allocation of the current project and then save all new entries.

As described in the Used Tools of Proposed Methodology, several tools and frameworks were used in the development of this application, one of those being Maven, which has commands that can compile, build and package the application into a jar file, this can be obtained by running `mvn clean package` this will contain all the code and dependencies that it requires but also it has the possibility of being executed through the command line.

Since we have this possibility, it is possible to create a .bat file with a java command that starts the application which in turn will be used to run the program on a regular basis, the way this part is configured will depend on the operating system of the server where this jar will run.

In my case, on my local work environment I have Windows Operating System installed which has an application called ”Task Scheduler” where it is possible to create a scheduled process that runs a .bat file, and thus update the data automatically.

### 4.4.2 Timesheets

At InnoWave, as already mentioned, the problem of employees not filling out timesheets on time exists, which results in a problem for the Finance department. In this section, I intend to present a possible solution to this problem using my Employees_Charge_Codes table.

Currently a SQL process is triggered every week automatically which inserts in the fmr table from Orange the timesheets informations which are the existing holidays depending on the geographical area of the employee. Based on this process and the new Employees_Charge_Codes table, where there is an association between an employee and a charge code, as well as the dates on which it must be used, it will be possible to pre-fill the timesheets in an automatic way.

For this new process to be implemented, a procedure must be developed in SQL in which, for each employee, depending on the current week, the current charge code that must be used is found, information present in the Employees_Charge_Codes table, and this is inserted on the database.

Each employee can be allocated to more than one project in a given period of time or have an allocation in a project less than 100%, in these cases the hours entered in the timesheet must take into account these values so that it is close to reality.

This procedure will only pre-fill the timesheet, not submit it directly, for this it is necessary for each employee to validate that the pre-filled information is correct and then...
submit the timesheet in time.

This automation is currently not possible to implement due to the fact that the information regarding the dates on which a charge code must be used by an employee only exists in the excel sheets that each manager creates to know the overview of their employees.

### 4.5 Testing

To prove the functionality of all the features in the different parts of the system, several tests were made, all done manually by me, these will be described below as well as the results obtained in each one.

**Power BI**

The validations made on all dashboards consisted of:

1. Confirming in the database that the displayed values are indeed correct.
2. All filters have been tried and the information in the database has been validated again.
3. The data restriction depending on the manager was tested by directly changing my professional email in the database by the email of another employee and it was confirmed that in fact only the employees to which I had access at the time were displayed.
4. Finally, the functionality of clicking on the hyperlink and being redirected to the Employee Details screen in the case of the Employees Allocation Overview and Employees Per Project dashboards and being redirected to the Employee Abroad Details screen in the case of the Employees Abroad dashboard.

**Power Apps**

On the application, in addition of having validated all the information shown, taking into account the information contained in the database, 9 tests were performed which include all possible flows in the application and thus proved that the application does what it expected. Next, the tests are listed, which steps were taken to complete the test and, finally the result obtained.

**Test 1**

Associate a new charge code to an employee with the Mixed work model, a duration of 6 months, where the first 3 months the employee will work in Nearshore mode and the last 3 months will be On site and requires an accommodation.
Steps

1. On the Employee Details screen click on the “Add Project” button.
2. Select a charge code that is not associated with the employee.
3. Choose the “Mixed” work model.
4. Set the start date and end date with the difference of 6 months.
5. Define the employee’s allocation with the value 100.
6. Click on the ”Specify date” button.
7. Select the ”Nearshore” working model.
8. Set end date 3 months after start date.
9. Click on the ”Add Another Date” button.
10. Select the ”On site” work model.
11. Leave the end date with the default value, which will be the end date entered in the fourth step.
12. Click on the toggle to mention that an accommodation is needed.
13. Click on the ”Save” button.
14. Since an accommodation is needed and the toggle was turned on, an accommodation request is already pre-filled with the charge code, start and end date.
15. Define the city of the accommodation.
16. Add notes if necessary.
17. Click on the ”Additional Information” button.
18. Add required CC emails.
19. Add extra information about the accommodation request.
20. Click on the ”Send Email” button.

Result  In the table with the history of the projects, present in the Employee Details screen, 2 entries will be added, one will have the work model Nearshore and the other will have On site, both with the dates and allocation specified by the manager. Since an accommodation was requested, in the table of accommodations, which is on the Employee Abroad Details screen, there will be a new entry with the data entered and an email with this information will be sent to the Office.
Test 2

Change the end date of a current project associated with an employee to a more recent date.

Steps

1. On the projects table, present on the Employee Details screen, click in the “Edit” column of the employee’s current project, this action is valid only for projects with "Current” or "Future” status.
2. Change only the end date to a more recent date.
3. Click on the ”Save” button.

Result  In the table with the history of the projects, the end date of the project will be changed to the date introduced by me.

Test 3

Change employee allocation on current project to 50%.

Steps

1. On the projects table, present on the Employee Details screen, click in the “Edit” column of the employee’s current project, this action is valid only for projects with "Current” or "Future” status.
2. Since it is an allocation change, change the start date to when the employee will have this allocation on the project.
3. End date in this scenario will remain the same as the working model.
4. Click on the ”Save” button.

Result  In the table with the history of the projects, the end date of the project that I selected will be changed to the new start date defined and subtracting one day, a new entry is added in the table with the start and end dates defined by me with the allocation of 50%.

Test 4

Remove the association between an employee and a project.
Steps

1. In the table of projects, present on the Employee Details screen, click on the column "Remove" from the employee’s future project, an action valid only for projects with the status of "Future".

Result   In the table with the history of the projects, the entry that I clicked on will be removed.

Test 5

Extend the duration of an abroad employee for 3 months.

Steps

1. In the accommodations table, present on the Employee Abroad Details screen, click on the “Edit” column of the employee’s current accommodation, this action is valid only for accommodations with the “Current” or “Future” status.

2. Change end date to 3 months in the future.

3. Click ”Additional Information” button.

4. Add required CC emails.

5. Add extra information about request to change accommodation.

6. Click on the ”Send Email” button.

Result A new entry will be added on the accommodation table with the start date being the previous end date and the new end date is the value entered by me, an email with all the information entered in the application is also sent to the Office department to request the changes.

Test 6

Request the cancellation of an accommodation.

Steps

1. In the accommodations table, present on the Employee Abroad Details screen, click on the ”Cancel” column of the employee’s future accommodation, this action is valid only for accommodations with the “Future” status.

2. Click ”Additional Information” button.
3. Add required CC emails.

4. Add extra information about the accommodation cancellation request.

5. Click on the "Send Email" button.

**Result**  The accommodation selected by me will be removed from the accommodation table, and an email with all the information entered in the application will be sent to the Office department requesting cancellation.

---

**Test 7**

Request a return flight

**Steps**

1. On the Employee Abroad Details screen click on the "Request new flight order" button.

2. Select the charge code that should be associated with this order.

3. Set the departure city of the outbound flight.

4. Set the destination city for the outbound flight.

5. Set the day and minimum time for the outbound flight.

6. Click on the toggle to mention that a return flight is needed.

7. Click on the "Return Flight" button.

8. The charge code, city of origin and destination are automatically filled based on the data previously entered, but can be changed as needed.

9. Set the day and minimum time for the return flight.

10. Click on the "Additional Information" button.

11. Add required CC emails.

12. Add extra information about ordering flight tickets.

13. Click on the "Send Email" button.

**Result**  In the flights table, which is on the Employee Abroad Details screen, there will be two new entries with the data provided, one of which will be for the outbound flight and the other for the return flight, and an email with their information is sent to the Office.
**Test 8**

Requesting a flight time change.

**Steps**

1. In the flights table, present on the Employee Abroad Details screen, click on the "Edit" column of the employee’s flight ticket, this action is valid only for flight tickets with "Current" or "Future" status.

2. Change flight date and time to desired values.

3. Click "Additional Information" button.

4. Add required CC emails.

5. Add extra information about flight ticket change request.

6. Click on the "Send Email" button.

**Result**  The flights date and time will be updated in the flights table with the values entered by me, an email with all the information entered in the application is also sent to the Office department to request the changes.

**Test 9**

Request cancellation of an flight ticket.

**Steps**

1. In the flights table, present on the Employee Abroad Details screen, click on the "Cancel" column of the employee’s future flight ticket, this action is valid only for flight tickets with the "Future" status.

2. Click "Additional Information" button.

3. Add required CC emails.

4. Add extra information about flight ticket cancellation request.

5. Click on the "Send Email" button.

**Result**  The flight ticket selected by me will be removed from the flights table, and an email with all the information entered in the application will be sent to the Office department requesting cancellation.
**Power Automate**

To validate all the created flows, I used the built-in Power Automate functionality to run the flow manually, so I triggered the flow manually using my email in order to receive the email generated by the flow and validate if the information and its formatting was correct.

### 4.6 Work Analysis

After all the work has been described, all functional and non-functional requirements of the system were met or not. For functional requirements, it will be indicated where on the system the required information can be found, in the case of non-functional requirements, it will be indicated how the requirement is met.

**Functional Requirements**

1. Show the allocation of an employee within a manager-specified time frame.
   
   (a) Information available on the Employees Allocation Overview dashboard.
   
   (b) There are filters that change the displayed time interval according to the manager’s preferences.

2. Show each employee’s project history.
   
   (a) Information available on the Employees Per Project dashboard.

3. Show the end date of an employee on a project.
   
   (a) Information available on the Employees Allocation Overview dashboard.

4. Show employees abroad as well as information about their accommodation and airplane tickets.
   
   (a) Information available on the Employees Abroad dashboard.

5. Show the following summary numbers of an SBU: Employees without a project, Partially allocated employees, Fully allocated employees, Employees close to being without a project, Employees whose accommodation is close to the end date, Airplane tickets required to request the Office and Total number of employees the manager has access to as well as the Total number of employees abroad.
   
   (a) Information available on Employees Overview KPIs dashboard.

6. Show the following information about charge codes: End date, Customer name and Employees using them.
7. Show the total number of clients and projects within a manager-specified time frame.
   (a) Information available on the Projects Overview dashboard.

8. Add, edit and remove the association of an employee to a project, where the charge code, work model, start date, end date and allocation are specified.
   (a) Actions available on the application’s Employee Details screen.

9. Add, edit and remove an accommodation request to the Office, where the charge code, location, start date, end date and notes if needed, are specified.
   (a) Actions available on the application’s Employee Abroad Details screen.

10. Add, edit and remove the request for a flight ticket to the Office, where the charge code, city of departure, city of arrival, minimum flight date and time and notes if needed, are specified.
    (a) Actions available on the application’s Employee Abroad Details screen.

11. Notify an employee of its new charge code and on what dates he should use it, as well as any changes made to its current charge code.
    (a) Actions available on the application’s Employee Details screen when the manager clicks on the check box during changes.

12. Notify managers about upcoming end dates for an employee on a project, accommodation, flight ticket and charge code.
    (a) Automatic email flows created in Power Automate.

**Non Functional Requirements**

1. The system should run on any platform.
   (a) All platforms of Power Platform used in the system can be accessed through a computer, tablet or smartphone.

2. System response times must not be long, it must be a responsive system.
   (a) In the Employees Allocation Overview and Employees Abroad dashboards, when filters are changed in order to search for information, the waiting time is sometimes a little high, taking a few seconds until the matrix is fully rendered.
(b) In the application, its responsiveness is not immediate, but it is not that time-consuming in actions such as adding or changing information.

3. The system interface must be user-friendly.

   (a) The design of each part of the system always took into account the visual aspect, which is one of the essential points of the system, as the graphical interface will be the point of contact between the manager and the system, so it was designed to be as easy and intuitive as possible.

4. The information shown must take the manager into account.

   (a) Throughout the system, the manager only sees the information he has access to.

5. The system must support several different managers at the same time.

   (a) Power Bi and Power Apps services support multiple managers interacting with them at the same time.

6. Time to learn how to use the system must be short.

   (a) Both the dashboards and the application were designed to be as interactive as possible so that managers quickly know how to use the system.

7. The system must be built in such a way that it will be relatively easy to add new modules in the future.

   (a) The system enhancement is easy and it doesn’t need a lot of effort to do it, new tables can be easily added to save new information, new dashboards to display it and new screens in the application to be able to edit them.

8. Easy implementation of new modules by other employees.

   (a) All platforms in the Power Platform are low-code which means that people who are not directly connected to a programming language will be able, with little effort, to develop new modules for the system.

9. The system must be developed in a sandbox environment.

   (a) The system was implemented in a completely isolated way and can be used without interfering with data from the other platforms.
Chapter 5

Conclusion

This project had the objective of developing a system that would help to centralize information existing in several different systems of InnoWave that do not communicate with each other. Currently the centralization of information is done by managers manually and subject to human error, my system was planned and developed always trying to automate as many tasks as possible, in order to be of value to the managers.

All the work described in this thesis was prepared and developed by me, this implies that I had to perform several tasks adjacent to the construction of a system from scratch, having started by collecting requirements with several managers, in order to know what information is needed, then all the analysis, planning and implementation of the 3 modules of this thesis was done, they are: Management of SBU, Projects Overview and Automated Processes.

For the development of this thesis, Microsoft’s Power Platform was used, it is composed of 4 components: Power BI where several dashboards were created, Power Apps used to create the application, Power Automate to create automated flows for sending emails and Power Agents which was not used in the system.

During this thesis several concepts and methodologies acquired in the master were used, they were consolidated and put into practice in a business environment which was beneficial to me.

In the analysis made on section Work Analysis of the chapter Work I presented all the functional and non-functional requirements and as it can be seen, all of the functional requirements were met which means that the system can solve the problems that currently exist and meet the needs of the managers.

In relation to the non-functional requirements, only one was partially fulfilled, which was: System response times must not be long, it must be a responsive system. In the dashboards that have matrixes, sometimes, the rendering time is a little long due to the calculations that are done behind the scenes, in the rest of the system the response times are better, fulfilling this requirement.

To conclude I can say that this thesis was completed successfully due to the facts
previously mentioned and also the comments made by the managers when the presenta-
tion took place, everyone gave positive feedback and saw great potential in this system
and the main objective of this thesis was also met, which was to centralize information
automatically for the managers.

5.1 Future Work

Taking into account all the content and the system developed in this thesis the following
points indicate the next possible steps to take in the future:

- **Integration with the tools** - Since in this thesis the system was developed in a
  sandbox environment, the next step will be to make the integration with the different
  existing platforms in order to obtain the live version of the system. This integration
  should not be very difficult to do due to the fact that the tables used are based on
  the real ones.

- **Database** - The integration with the various platforms can be done in two ways:
  
  1. All the system components connect directly to the different platforms, being
     necessary to work a little on the collected data in order to obtain the data
     format that I am currently using.
  
  2. Only Power BI connects to the different platforms, works the data and writes
     it to my data model, this way, in the rest of the components of my system it
     will not be necessary to make changes because the tables will not change.

- **Timesheet pre-fill automation** - Implement timesheet pre-fill for all employees
  based on the proposal presented in section [Automated processes].

- **Display vacations** - Currently on the Employees Allocation Overview dashboard
  the employees vacation are not displayed it only shows if the employee is on a
  project or not, a further improvement would be to also display the employee vaca-
  tions.

- **Improve performance** - Since the matrixes in the Power BI dashboards take some
  time to render due to the calculations made on the measures, they can be revised to
  improve their performance.
Appendix A

System dashboards

A.1 Employees Allocation Overview with date filters

Figure A.1: Employees Allocation Overview with date filters
A.2 Employees Allocation Overview with information filters

Figure A.2: Employees Allocation Overview with information filters

A.3 Employees Allocation Overview with matrix filters

Figure A.3: Employees Allocation Overview with matrix filters
A.4 Employees Per Project with information filters

Figure A.4: Employees Per Project with information filters

A.5 Employees Per Project with matrix filters

Figure A.5: Employees Per Project with matrix filters
A.6 Employees Abroad with date filters

Figure A.6: Employees Abroad with date filters

A.7 Employees Abroad with matrix filters

Figure A.7: Employees Abroad with matrix filters
Appendix A. System dashboards

A.8 Employees Overview KPIs with date filters

Figure A.8: Employees Overview KPIs with date filters

A.9 Projects Overview with date filters

Figure A.9: Projects Overview with date filters
A.10 Projects Overview with information filters

Figure A.10: Projects Overview with information filters

A.11 Projects Overview with matrix filters

Figure A.11: Projects Overview with matrix filters
Appendix B

System application

B.1 Employees with work model filters

Figure B.1: Employees with work model filters
B.2 Employees with SBU filters

![Employee List]

Figure B.2: Employees with SBU filters

B.3 Employee Details edit notes

![Employee Details]

Figure B.3: Employee Details edit notes
B.4 Employee Details project allocation details

![Employee Details project allocation details](image)

Figure B.4: Employee Details project allocation details

B.5 Employee Details new project allocation

![Employee Details new project allocation](image)

Figure B.5: Employee Details new project allocation
B.6  Employee Details edit project allocation

Figure B.6: Employee Details edit project allocation

B.7  Employee Abroad Details accommodation details

Figure B.7: Employee Abroad Details accommodation details
Appendix B. System application

B.8 Employee Abroad Details request new accommodation

Figure B.8: Employee Abroad Details request new accommodation

B.9 Employee Abroad Details request an accommodation update

Figure B.9: Employee Abroad Details request an accommodation update
Appendix B. System application

B.10 Employee Abroad Details request a new / update / cancellation accommodation extra information

Figure B.10: Employee Abroad Details request a new / update / cancellation accommodation extra information

B.11 Employee Abroad Details flight ticket details

Figure B.11: Employee Abroad Details flight ticket details
Appendix B. System application

B.12 Employee Abroad Details request new one-way flight ticket

![Employee Abroad Details request new one-way flight ticket](image)

**Figure B.12:** Employee Abroad Details request new one-way flight ticket

B.13 Employee Abroad Details request new return flight ticket

![Employee Abroad Details request new return flight ticket](image)

**Figure B.13:** Employee Abroad Details request new return flight ticket
Appendix B. System application

B.14 Employee Abroad Details request a flight ticket update

Figure B.14: Employee Abroad Details request a flight ticket update

B.15 Employee Abroad Details request a new / update / cancellation flight ticket extra information

Figure B.15: Employee Abroad Details request a new / update / cancellation flight ticket extra information
B.16 Office Input return flight update

Figure B.16: Office Input return flight update

B.17 Office Input accommodation update

Figure B.17: Office Input accommodation update
Bibliography


