Universidade do Minho
Escola de Engenharia
Departamento de Informática

Maria Moutinho Figueiredo da Silva

Improve human resources management via UX/UI

October 2022

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Maria Moutinho Figueiredo da Silva

Improve human resources management via UX/UI

Master dissertation
Integrated Master's in Informatics Engineering

Dissertation supervised by
José Creissac Campos

October 2022

ACKNOWLEDGEMENTS

I would like to thank the following people for helping me finalize the project. Without them this project would not have been possible.

I would like to express my gratitude to my supervisor, Prof. José Creissac Campos, who patiently read and revised this thesis countless times, and always went out of his way to gather resources that could help. I truly could not have asked for a better supervisor, thank you for your time and constant support.

I am also extremely grateful to my VILT supervisors, Ana Anjo and Didier Alves. Without your support I could never have done this. Thank you for all the resources, meetings and discussions that led to our final solution.

I am also thankful to my many coworkers and friends at VILT who helped me even when their work was not related to mine, thank you for your kindness and support. A special thanks to Rita Soares, who motivated me when I felt like I was stuck, and who invited me to this amazing team.

I am deeply grateful for my boyfriend, who kept pushing me when I didn't think I could do it. To my friends who were also writing their own dissertations and still gave me support and empathy, we did it! To my mom, my dad, my brother, and my grandma, for believing in me and always giving me the best, thank you so much.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity.

I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

ABSTRACT

Resource allocation is critical to optimize billable allocation and avoid allocation conflicts. Tools are needed to correctly book and share allocations during the normal operation of a company.

Zeus is an intranet built in-house by VILT, a digital solutions company, which is used to connect every employee in the company, containing important personal and shared information, as well as useful services. Starting from scratch, it was developed according to the needs of VILT's staff, making it a fully customized intranet.

One of the many features in Zeus that was requested by VILT's Professional Services Managers is the allocation map page. This feature allows managers to manage multiple human resources and their projects. Each manager has their own team to manage, and they currently face some limitations related to the resource allocation management process. These limitations are noticeable to the managers, resulting in a lack of adherence from some of them, which resulted in a divergence of the methods used in resource allocation management inside the company. There is a need to create an improved solution that takes into account: the managers' insights and ideas, inspiration from existing software, and inspiration from the managers' alternative solutions. The main goal for the new solution is having an easy view of who is available and when, as well as the option to easily manage that information.

With this dissertation, we want to study the UX/UI Design process and different resource allocation solution, and use it to design a resource allocation solution for VILT.

KEYWORDS VILT, UX/UI, resource allocation management, project management.

RESUMO

A alocação de recursos é fundamental para otimizar a alocação faturável e evitar conflitos de alocação. São necessárias ferramentas para registar e partilhar corretamente alocações no funcionamento normal de uma empresa.

O Zeus é uma intranet construída internamente pela VILT, uma empresa de soluções digitais, e é usado para conectar todos os funcionários da empresa, contendo informações importantes pessoais e partilhadas, bem como serviços. Foi desenvolvido de acordo com as necessidades da equipa da VILT, tornando-se uma intranet totalmente personalizada.

Uma das características do Zeus que foi solicitada pelos *Professional Services Managers* da VILT foi a página do mapa de alocação. Este recurso permite aos *Managers* gerir múltiplos recursos humanos e os seus projetos. Cada *Manager* tem sua própria equipa para gerir, e estes atualmente enfrentam algumas limitações relacionadas ao processo de gestão de alocação de recursos. Estas limitações são visíveis para os *Managers*, resultando na falta de adesão de alguns deles, o que resultou numa divergência dos métodos utilizados na gestão de alocação de recursos dentro da empresa. É necessário criar uma solução melhorada que tenha em conta: as visões e ideias dos *Managers*, tendo inspiração no software existente e nas soluções alternativas usadas pelos *Managers*. O objetivo principal para a nova solução é ter uma visão fácil de quem está disponível e quando, bem como a opção de gerir facilmente essa informação.

Com esta dissertação, queremos estudar o processo de Design UX/UI e a gestão de alocação de múltiplos recursos, e usá-la para criar o design de uma solução de alocação de recursos para a VILT.

PALAVRAS-CHAVE VILT, UX/UI, gestão de alocação de recursos, gestão de projetos.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

1.1 CONTEXT

Resource allocation is critical to optimize billable allocation and avoid allocation conflicts. Tools are needed to correctly book and share allocations during the normal operation of a company.

Zeus is an intranet built in-house by VILT, which is used to connect every employee in the company, containing important personal and shared information, as well as useful services. VILT is a company focused in digital transformation, helping enterprises all over the world to continuously improve their customers experience and to become more efficient operationally through the adoption of digital solutions, full data integration and advanced analytics.

Being an in-house project, Zeus was developed according to the needs of VILT's staff, making it a fully customized intranet. Besides individual users, several areas benefit from it differently: Sales uses it to manage the opportunities, the Human Capital department to manage the people, Project Managers to manage the projects and, of particular interest in this case, manage resource allocation, etc.

1.2 MOTIVATION

One of the many features in Zeus that was requested by VILT's Professional Services Managers is the allocation map page. This is a feature that allows managers to manage multiple human resources and their projects. Each manager has their own team to manage, and having a way to organize and track the related information is necessary.

This feature already exists in Zeus. It allows managers (users with permission to access this tool) to view a set of employees of their choice, and their time allocations in each project in a personalized *gantt* chart, as well as creating, editing and deleting those time allocations. However, the way it was created contains some limitations related to the resource allocation management process. These limitations are noticeable to the managers, resulting in a lack of adherence from some of them, which resulted in a divergence of the methods used in resource allocation management inside the company. Currently, the time allocation of each team is being managed through different methods by each manager, depending on their preferences.

There is a need to run a new study that takes into consideration the managers' insights and ideas, inspiration from existing software, and inspiration from the managers' alternative solutions, in order to design an improved

solution. The main goal for the new solution is having an easy view of who is available and when, as well as the option to easily manage that information.

1.3 OBJECTIVES

With this dissertation, we want to study the UX/UI Design process and different resource allocation solutions. The objective is to use these concepts to design a new resource allocation solution for VILT. In order to achieve this objective, we need to complete 3 smaller objectives: The first objective is to do some user research and collect the managers' ideas about what they like and dislike in the current allocation map, as well as ideas for the new one. The second objective is creating prototypes and studying the managers' feedback until we reach a final solution that is in a good state for future implementation. The third and last objective will be gathering feedback about the results and discussing it for future learning, as well as discussing future steps for when the implementation stage is realized.

1.4 DOCUMENT STRUCTURE

The dissertation has the following structure:

Chapter 1: Introduction, where the theme is explained, as well as its context, motivation and objectives.

Chapter 2: User Experience (UX), where important user experience concepts and stages will be studied and explained.

Chapter 3: Resource Allocation, where relevant project management and resource allocation concepts will be explained, along with an introduction to the problem.

Chapter 4: Creating a Solution, where we will finally use the knowledge we gathered in the previous chapters to create a final prototype.

Chapter 5: Conclusion, where we discuss the results and propose some lines for future work.

2

USER EXPERIENCE (UX)

User experience (UX) is defined in ISO 9241–210 as "A person's perceptions and responses that result from the use and/or anticipated use of a product, system or service" [1].

User experience design (UXD) is the process used to create products that provide meaningful and relevant experiences to users. This involves the design of the entire process of acquiring and integrating the product, as well as aspects of branding, design, usability and function [2].

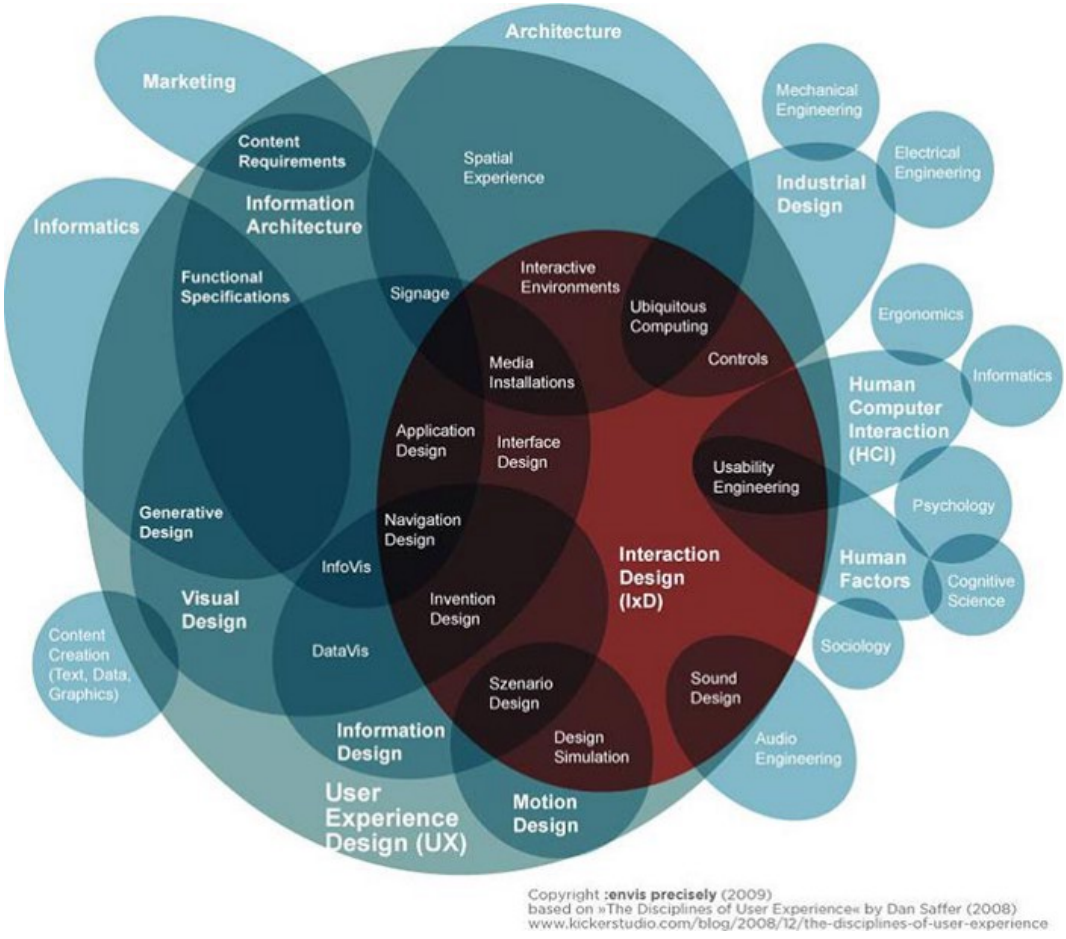

Figure 1: Image from Human Computer Interaction — brief intro [3]

User experience is a term that contains lots of subsets (see Figure 1). All of these are useful in the creation of a new product, but we can focus on specific ones depending on the kind of product.

One of the components of UX is HCI (Human-Computer Interaction). HCI focuses mainly on the design of the interaction between the user and the computer. HCI is more about the user's experience of interacting with the computer, while UX is about the user's whole experience of interacting with the product or service.

In the context of this work, we will give special focus to HCI. Since the context is already well defined, and the main focus will be the existing users, and not future buyers, learning about HCI will help us make a better analysis of which components work the best to allow a good interaction between the user and the computer (i.e. interactions, keyboard, screen, mouse, etc).

The UX approach usually starts with understanding the users and the problem's requirements. After that, researching competitors, trends and related guidelines. Then comes the creative process, brainstorming ideas, and designing prototypes. When this is done, the solution can be implemented, and in the end feedback is collected and used as inspiration for improvement.

The HCI approach is similar but includes less stages: understanding the user, designing, and evaluating. After having an analysis of the users, design guidelines are followed to create the initial design concepts. After this is done, some usability tests can be made with low-fidelity mock-ups or prototypes. This helps achieve a better understanding of the user's needs and struggles. This testing process should be iterated many times.

While both UX and HCI concepts will be addressed, this chapter will be divided using the HCI stages to simplify reading.

2.1 UNDERSTANDING THE USER

In this section, the main focus will be the user. The section is divided in two parts: first, we will analyse psychological factors, and later we will research methods and techniques to better understand the user.

2.1.1 *Cognition - Human Behaviour*

Before trying to understand the specific final users, we want to study the human mind and how that can help us in the creation of the solution. For this, we resort to [4] to present concepts explaining what catches people's attention more easily, short and long term memory, and how to choose the best kind of software for our users' level of knowledge.

Selective Attention

The human mind usually uses four factors to select what gets attention and what is ignored:

- **Salience:** Salient stimulus are used to signal important events, like alarms and alerts. Some events may go unnoticed without this, even if they are important.
- **Effort:** Selective attention may choose not to pay attention to something if it is effortful.

- **Expectancy:** People are more prone to looking at places where they expect to find information.
- **Value:** People will pay more attention to something if it is valuable to look at, or if it might be costly to not see it.

Some guidelines that can be followed to support perception include:

1. Increase visible legibility and pay special attention to confusion caused by similar messages in different contexts.
2. Use familiar representations, like familiar fonts, meaningful icons, and words rather than abbreviations.
3. When familiar representations are unavailable (unfamiliar symbology), try to provide the best opportunities for guessing: use a smaller vocabulary, and create context (like a phrase, instead of just using keywords).

Working Memory

There are two types of memory storage in the human brain:

- Working memory, or short-term memory: Relatively transient and limited to holding a small amount of information. It is the temporary store that keeps information active while we are using it or until we use it.
- Long-term memory: Involves the storage of information after it is no longer active in working memory and the retrieval of the information at a later point in time.

Working memory is limited by:

- **Capacity:** How much information can be kept.
- **Time:** How long the information can be kept.
- **Confusability and Similarity:** How similar the information is to other elements in working memory.
- **Attention and Similarity:** How much attention is needed to keep the information.

Considering these limitations, we can try to make the solution easier to the user by doing the following:

1. Reduce working memory load: the time and size of the information that need to be retained should be reduced as much as possible.
2. When there are sequential tasks, provide a visual reminder of the previous completed steps, so that the user does not need to go back or start again because of forgetting what has been done.
3. Reduce confusability, by building physical distinctions of the information that needs to be retained.
4. Reduce the use of negation where possible, as this adds up to the size of the information. If the negation ends up being forgotten, the instruction becomes the opposite of the intended.

Long-Term Memory

When information is stored in long-term memory, this is called learning. And to help with the learning process, training features can be created.

In long-term memory, it is important to know which factors make it easier to retrieve certain pieces of information. These are:

- **Strength:** determined by the frequency and recency of its use. When a piece of information is accessed everyday, it is less likely to be forgotten anytime soon, than another piece of information that is only accessed once a month. So, when procedures are not supposed to be done very frequently, they should be supported by visual checklists or guides rather than just rely on the user's memory alone.
- **Associations:** The items in long-term memory are linked with other related items through associations. Each of these associations between items has a strength, just as each item does. If time passes and the associations are not repeated, they will become weaker. The more associations there are to an item, the more likely will be that it can be found in long-term memory and retrieved.

Then, forgetting usually happens because of:

1. Weak strength due to low frequency or recency.
2. Weak or few associations with other information.
3. Interfering associations.

Speaking about associations, it is important that the structure of the created database is compatible with the organization of the user's semantic network. Items that are close together in the user's semantic network should be close together in the database representation of the information.

We can separate long-term memory retrieval into two forms:

- **Recall:** the user must retrieve the required information from memory.
- **Recognition:** the user is given a perceptual cue given in the environment, that triggers an association with the desired information.

Recall is lost faster than recognition. In HCI, command languages require recall of the specific commands, and menus allow visual recognition of the command to be clicked. These should be considered with the type of users that will use the software.

User vs System

An important thing to remember is that the greater the number of features, the more difficult it will be to make the solution usable and user-friendly. If there is no attention to this, there will be a big gap between the demands placed on the user and the user's capabilities.

To make this more balanced, we can focus on the following:

1. **Frequency of use** - people who will be using the software frequently are more willing to invest some time learning. Besides this, if a task is frequently performed, it will be more easily remembered.

2. **Mandatory versus discretionary use** - if the use of the program is mandatory or it has high frequency of use, there should be a focus on ease of use; if it is optional or has low frequency of use, ease of learning and remembering should be prioritized.

3. **Knowledge level of the user** - this will influence the user's ability to memorize procedures or easily learn them. If the user is a novice in the subject, software relying on a GUI with menus, icons and short instructions will be more intuitive, since this only requires memory for recognition and not recall. However, if the user has advanced knowledge on the subject, using command-line interaction may allow the user more efficiency in doing the tasks.

2.1.2 *User Research*

User research can be separated into: **primary research** - feedback generated during previous projects, and **secondary research** - consumer market research reports [5].

Information can be **quantitative** - hard statistical numbers about the size and composition of target user groups, or **qualitative** - information about what that user group buys or consumes and what their lifestyle is like. Quantitative information enables a design team to put physical dimensions to a target market. Qualitative information allows the design team to understand why people respond to stimuli or not. It is typically obtained via face-to-face interviews where participants talk about their experiences and preferences

Examples of surveys to obtain information of both types include:

- **Statistical surveys** - Collects quantitative information from numbers
- **Sampling** - Collects information from a population sample
- **Opinion polls** - Assess public opinion
- **Paid statistical surveys** - Rewards participants for the answers
- **Questionnaires** - Contains a set of questions
- **Omnibus surveys** - Asks questions in a regular monthly survey

To perform a study, a sample group is usually created by selecting a collection of 5 to 10 people who can be used for one-to-one interviews, questionnaires and focus groups. This sample should be selected by first determining the most important attributes that define the target group.

As the goal is to design a system for mandatory use done by existing users, with no need for sales, we will focus on primary research of qualitative information.

2.2 DESIGNING

Now that we have useful information to better understand the user, we need methods to design a usable and intuitive solution that is more than just aesthetically pleasing. For this, we will focus on brainstorming and prototyping methods, and then discuss some additional guidelines that should be considered.

2.2.1 *Brainstorm and Prototypes*

The first step towards creating a solution will be performing brainstorming sessions. This brainstorming process starts by selecting group participants that will address the problem and raise questions or suggestions for the new solution.

We can choose two specific directions [5] in order to create new ideas from existing designs:

- **Divergence:** Moving away in different directions from a common point, this is, creating a solution that stands out.
- **Convergence:** Joining two or more designs toward a central solution or common ground.

Our solution will follow a convergent design, using multiple solutions as inspirations to create a new one.

After finishing the brainstorming process, the next step is creating a prototype. The main methods for prototyping digital solutions are sketching, or creating a digital mock-up.

A prototype gives the design team and the client the ability to better visualize a concept. Its purpose is to test specific aspects of a design solution, so that these can be evaluated. Prototypes can be created in two levels: *macro* or *micro*. At the *macro* level, the result will be a low-fidelity prototype, or a thumbnail sketch. At a micro level, the result is a high-fidelity prototype, or a more refined and detailed version, closer to the final design.

When a prototype is created, the solution should be presented to the client. This will generate feedback, which can be used for further improvement.

2.2.2 *Guidelines*

Usability

ISO 9241-11, defines usability as: The “extent to which a product can be used by specified users to achieve specified goals effectively, efficiently and with satisfaction in a specified context of use” [6, 7].

When thinking about usability, we should think about the following components [8]:

- **Learnability:** How easy is it for users to accomplish basic tasks the first time they encounter the design?
- **Efficiency:** Once users have learned the design, how quickly can they perform tasks?
- **Memorability:** When users return to the design after a period of not using it, how easily can they reestablish proficiency?

- **Errors:** How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- **Satisfaction:** How pleasant is it to use the design?

Some useful usability guidelines are [9]:

- **Match between system and real world** - speak language that is familiar to the user, follow real-world conventions, use familiar metaphors.
- **Consistency and standards** - Use color coding uniformly, group functions logically and consistently, conform to platform interface conventions.
- **Visibility of system status** - inform the user about what goes on, show what input has been received, indicate progress in task performance, provide feedback for all actions.
- **User control and freedom** - allow undo, cancel and redo actions, mark exits clearly, allow user to initiate/control actions.
- **Error prevention, recognition and recovery** - prevent errors from occurring, help users recognize, diagnose and recover from errors, use clear error messages.
- **Memory** - use "see-and-point" instead of "remember-and-type", provide lists of choices and picking from lists.
- **Flexibility and efficiency of use** - provide shortcuts, give user options to speed up frequent actions, make system efficient to use.
- **Simplicity and aesthetic integrity** - use a simple graphic design, use simple and natural dialog, show information in a natural and logical order.

These guidelines will help users by making the software easier to learn and use.

Dialog Style Guidelines

Because there is the possibility that the users will need to input information, or choose existing settings, options or functionalities, it is important to know what are the existing types of communication between the software and the user. For this, there are different types of dialog styles that can be used in the software. For a better use of each dialog style, we can follow the guidelines [4]:

- Menus: List of items from which to choose.
  - Gray out inactive menu items.
  - Create logical, distinctive, and mutually exclusive semantic categories.
  - Choose brief and consistent menu labels.

  - Order menu choices by convention, frequency of use, or order of use.
  - Use existing standards, such as: File, Edit, View that are common on most Windows applications.

- Fill-in forms: Provides blank spaces to enter information.
  - Use white space and separate logical groups.
  - Allow forward and backward movement.
  - Keep interdependent items on the same screen.
  - Indicate whether fields are optional.

- Question/answer: Provides one question at a time to type answer in a field.
  - Use visual cues and white space to clearly distinguish the questions and the input area.
  - Use clear and simple language.
  - Allow flexible navigation.
  - Minimize typing requirements.

- Command languages: User types commands with specific syntax.
  - Make the syntax as natural and easy as possible.
  - Make the syntax consistent.
  - Avoid arbitrary use of punctuation
  - Use simple, consistent abbreviations.

- Function keys: Pressing special keys or combinations to give commands.
  - Reserve the use of function keys for generic, high-frequency, important functions.
  - Label keys clearly and distinctly.
  - Place high-use keys within easy reach of home row keys and keys with serious consequences in hard to reach positions and not next to other function keys.
  - Minimize the use of “qualifier” keys (alt, ctrl, command, etc.) which must be pressed on the keyboard in conjunction with another key.

- Direct manipulation: Perform actions directly on visible objects.
  - Minimize semantic distance between user goals and required input actions.
  - Choose a consistent icon design scheme.
  - Accompany the icons with names if possible.

Design of User Support

Lastly, while the goal is to make the software so intuitive that no help is needed, this is an unrealistic goal to have. People will eventually need assistance or user support. For this, it is a good idea to include a variety of mechanisms such as software manuals, online help, tutorials, or other methods to help the user in case it is needed [4].

2.3 EVALUATING THE DESIGN

The last step of the whole creative process is evaluating the design. After designing a prototype, the result should be tested to check for possible improvements. This section addresses concepts that allow this evaluation to happen.

The main steps for user testing that will be followed are [8]:

1. Test the old design to identify the good parts that can be kept, and the bad parts.
2. Do some user research to know how users usually behave.
3. Make paper prototypes of new design ideas and test them.
4. Refine the best design ideas, and move from low-fidelity prototyping to high-fidelity representations, testing them.
5. Inspect the design relative to the requirements and guidelines established in previous research.
6. Once the final design is implemented, test it again, to make sure there are not any forgotten usability problems.

In this process, the main method for studying usability is user testing. This concept can be divided into 3 steps, and will be done for each new design that needs to be tested:

1. Choose representative users.
2. Ask the users to perform specific tasks.
3. Observe the users' responses and analyse what were the successes and difficulties in the user interface.

Users may respond to the new software in unpredictable ways, so there should be an effort to make the software fit the user and not the other way around. It should not be forgotten that the purpose of the software is to support the user in the necessary tasks, not to provide every feature that could be useful only once [4].

It is important to test users individually and to let them solve the tasks by themselves without help, so that the results are accurate and correctly represent what they would go through if that was the final product.

Usability tests are performed to simulate the interaction between the user and the software. The difficulties that the user faces are recorded to improve the software later.

During these tests, designers may ask users to think aloud during each task and answer questions. The goal is to find a prototype design that users like, learn easily, and can use to successfully perform tasks. Observation of users gives the designers insight into difficulties with controls, navigation, and so on. Once the interface has gone through several iterations, the prototype is then refined for more advanced usability testing [4].

2.4 CONCLUSION

In this chapter, we discussed the HCI approach of designing a solution, as well as related concepts that will help us in three different ways: understanding the users; designing a good solution; and evaluating the solution.

In the next chapter, resource allocation concepts will be researched, and Zeus will be introduced in better detail.

3

RESOURCE ALLOCATION

Resource allocation is critical to optimize billable allocation and avoid allocation conflicts. Tools are needed to correctly book and share allocations during the normal operation of a company.

The main focus in this chapter will be understanding the existing Zeus resource allocation management tool and gathering inspiration for the new solution.

3.1 ALLOCATION MAP

An allocation map is a tool that allows project managers to visualize and manage their team's time and projects. To better explain it, we can consider the following concepts:

Managers: A group of managers, which may include: Professional Services Managers, Software Factory Managers, Human Capital Manager, General Managers and Board Managers.

Allocation map: A table-style map in which columns are resources and lines are dates. Each cell represents a day of work (or a non-working day) for a single collaborator. Each cell should display the future allocation for each collaborator.

Allocation: An allocation represents the work that a person should do, in a period defined by the allocation period. An allocation may be partial or full. When partial, there should be more than one allocation for the same person, over the same date, in the same or in different projects. The work that a person needs to do is mainly a project created, or a custom task that is not associated to any project.

Allocation period: An allocation period is defined by a start date and an end date. When the start date and the end date spans over weekends, vacations, holidays or collaborator leaves, the allocation on these specific days is not accountable. Only if the start date or end date is explicitly defined on such non-working day shall that day be explicitly marked as allocated.

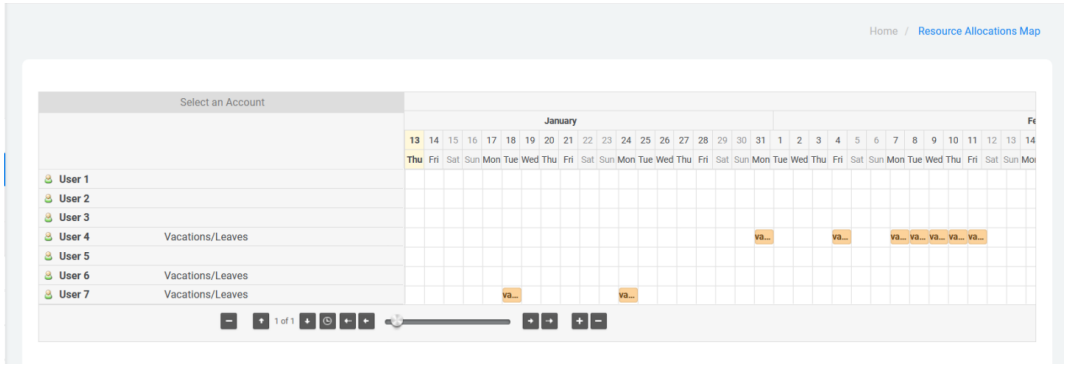

Figure 2: Zeus' current allocation map

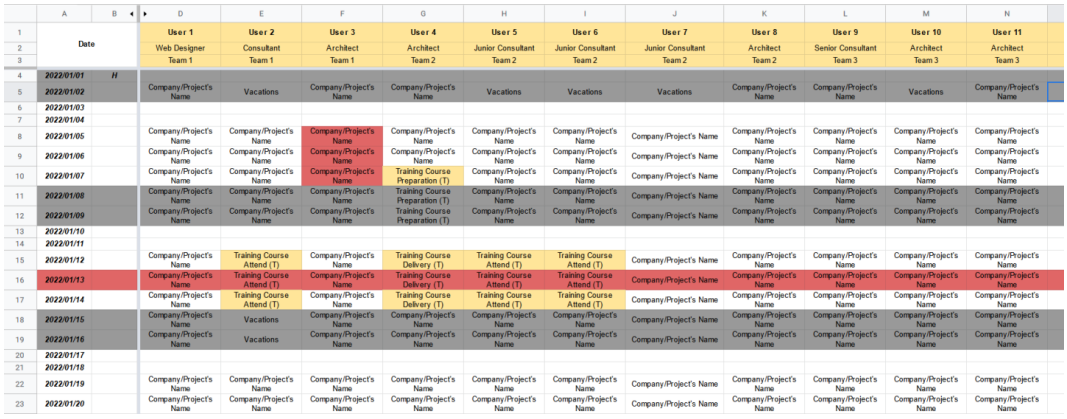

Figure 3: Spreadsheet used as replacement

3.2 ZEUS' ALLOCATION MAP

For intranets, usability is a matter of employee productivity. Thinking about this is important because if the new allocation map is difficult to use, people will waste time by trying to figure it out or trying to find a new and easier method to get their work done [8]. VILT considered that an allocation map was not just useful, but necessary to the Professional Services Managers in the management of their teams' time and project allocations. So, for this, an allocation map page was created in Zeus. However, no study was realized beforehand, and the solution was not entirely accepted by everyone. In the next chapter, it will be showed a survey that was answered by the managers. This is important to learn why the solution was not accepted. Currently, the time allocation of each team is being managed through different methods by each manager, depending on their preferences. Some write it manually in a notebook, some use the current Zeus' tool (see Figure 2), and some others use a spreadsheet (see Figure 3) to replace it.

In the future chapters, the creation of a new solution will be documented.

3.2.1 *Entities and Tables*

In order to create the best solution possible, we need first to understand what data we are going to work with. For this purpose, we can analyse the existing tables (see Figure 4) in the Zeus' database, specifically the ones related to the allocation map that already exists. It is important to know what information already exists in the database, since data already exists and it will have to be inserted in the new solution. It is important to understand if the new solution can be created using the existing data. Otherwise, if the existing tables or attributes are not enough, we can suggest changes for the future implementation team.

As we can see, in the current allocation map there are four tables related to it. There are users, projects, and opportunities stored in the database. An allocation can be created and stored, being associated with only one user and one type of project (project or opportunity).

3.3 EXISTING SOFTWARE

In order to understand what are some desirable features in resource allocation software, some research was made about what technologies already exist for the same purpose, that can be used as inspiration.

The following criteria [10] were analysed for each of them, and ranked from 1 (Unsatisfactory) to 5 (Outstanding):

- **User Interface (UI):** Does the design contain clear displays for the right resources when optimizing the team's workload? Is it appealing and easy to visualize?
- **Usability:** Is it intuitive and easy to learn?
- **Workload resource planning:** Is it possible to visualize many resources at a time, who is doing what, and who is available or unavailable?
- **Interactions:** Does it include useful interactions and shortcuts to quickly build a resource schedule template?
- **Timesheet integration:** Does it allow timesheet integration?
- **Resource allocation automation:** Is there a way to automate some of the resource allocation work?
- **Resource utilization alerts:** Does it inform the manager about overtime allocations and other problems?

The following observations were gathered based on a personal experience testing each software. Each software will also be shown later to a team of managers in order to collect more feedback (in the next chapter).

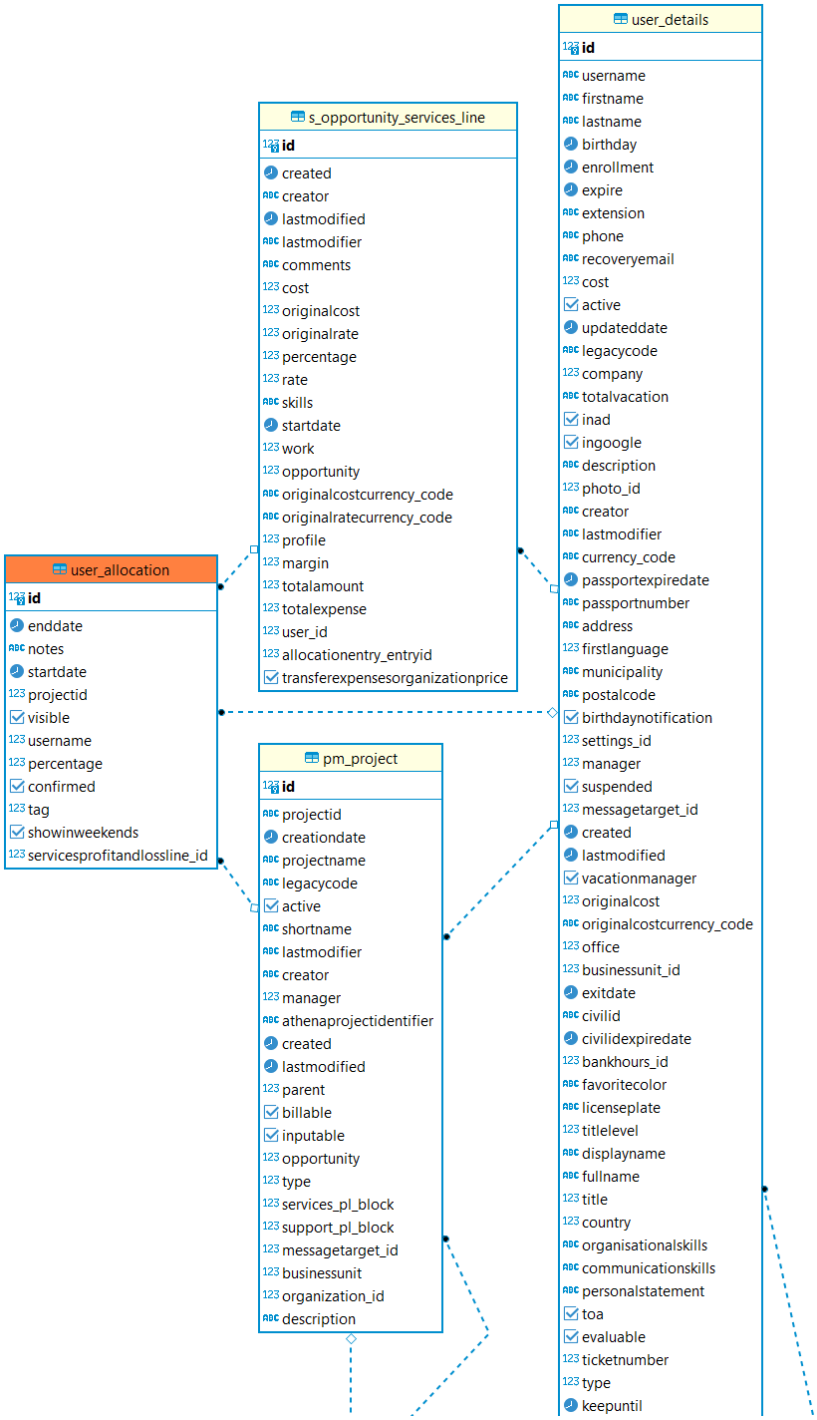

Figure 4: Tables related to the old allocation map

Figure 5: Runn - Overview, grouped by people

3.3.1 *Runn*

Runn[1] (see Figure 5) is easy to use and interpret. Several features may be useful, such as the possibility of tentative projects, view by different date ranges, a variety of search filters, and helpful interactions (for example, clicking on an empty date to create a new allocation, or clicking a block to split it). Finally, the data visualization graphs may inspire future improvements, since they can help managers a lot when it comes to analysing information.

3.3.2 *Mavenlink*

Mavenlink[2] (see Figure 6) offers the possibility to view availability filtering by time range statistics (see Figure 7), and data analysis reports. Besides these, some relevant features include a wide range of filters, view by day/week/month. Because of the increased number of features, managers may need more project management experience than in other software programs. Finally, the most unique feature about this software is the comparison between allocated and scheduled hours, i.e, planned and executed hours of work (see Figure 8).

3.3.3 *Float*

Although Float's[3] UI (see Figure 9) is simple and easy to use, it is difficult to interpret multiple resources at the same time. Unlike other programs, it does not allow views with different time ranges. One thing we can take from

1 http://www.runn.io
2 http://www.mavenlink.com
3 http://www.float.com

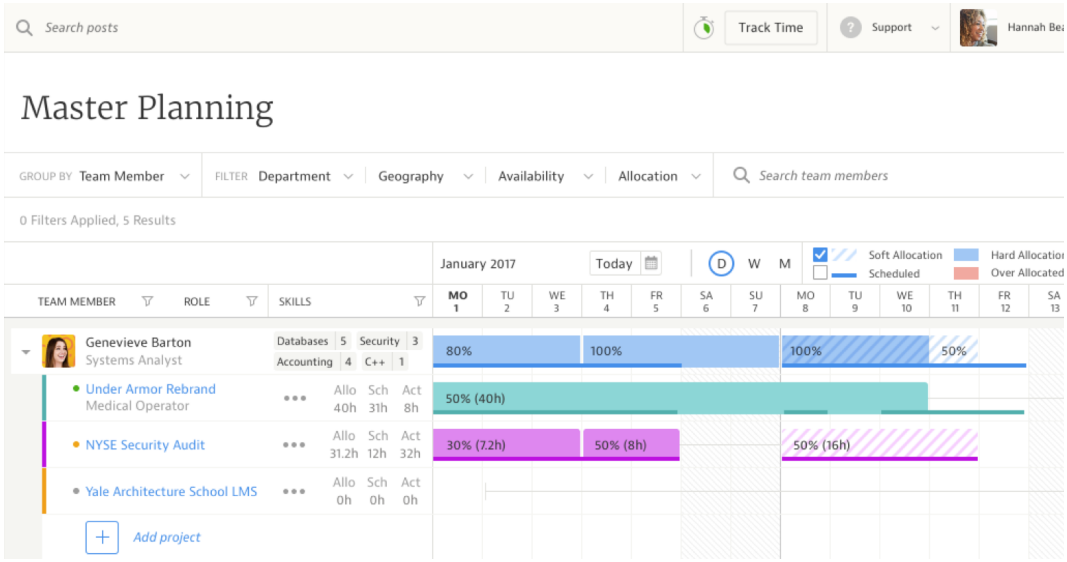

Figure 6: Mavenlink - Overview, grouped by team members

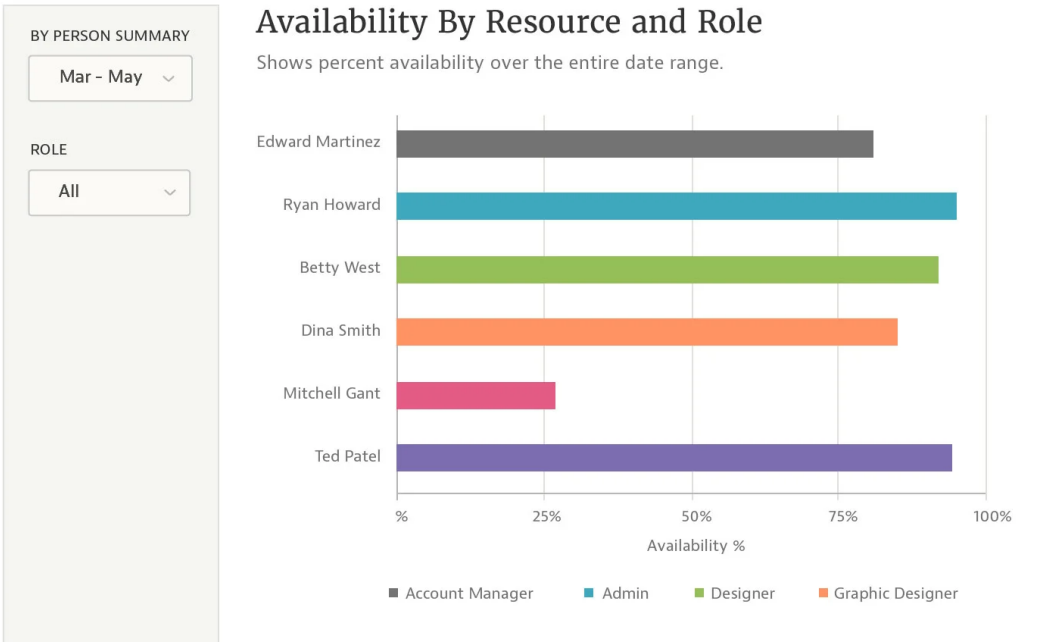

Figure 7: Mavenlink - Analysis of availability by role

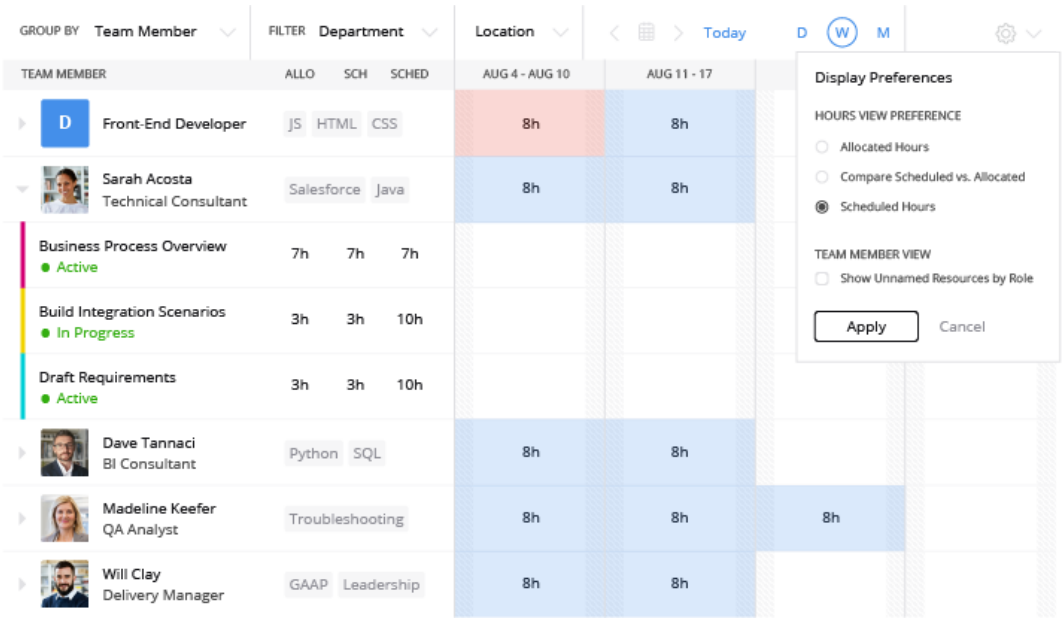

Figure 8: Mavenlink - Comparison between allocated and scheduled time

this choice is the diversity of interactions, for example: hold shift and drag to duplicate block, or drag ends of a block to expand/reduce it.

3.3.4 *Resource Guru*

Despite the visually simple UI, Resource Guru[4] (see Figure 10) is not very intuitive. The overall view gets too cluttered with multiple projects, showing only a limited number of people, making it difficult to interpret their availability. However, it has good interactions that we can get inspiration from, such as interaction animations, and clicking a block to split it in two.

3.3.5 *Elapseit*

While Elapseit[5] (see Figure 11) solves one of the problems found in some of the previous choices, i.e. it is able to show many rows at a time, the view is too cluttered and overwhelming to the manager. A good detail that we can take from this is the clear distinction between weekdays and weekends.

4 http://www.resourceguruapp.com
5 http://www.elapseit.com

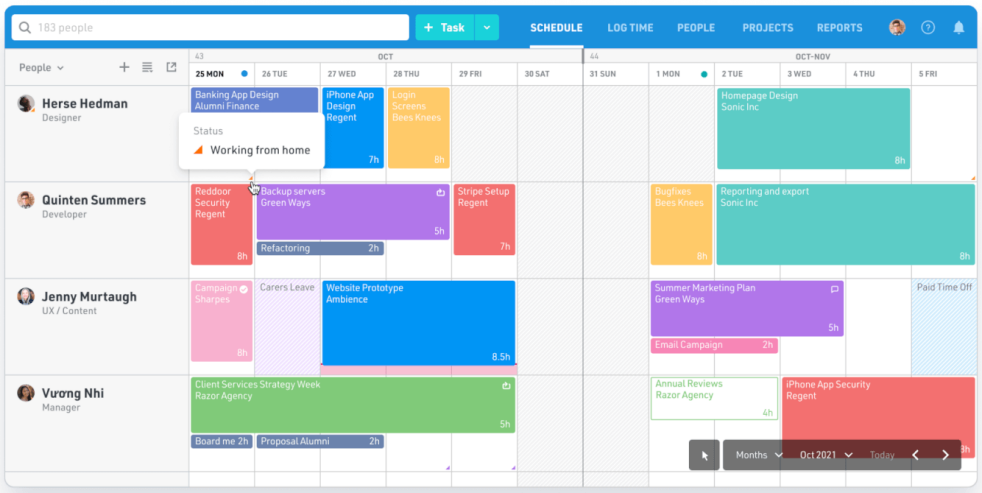

Figure 9: Float - Overview

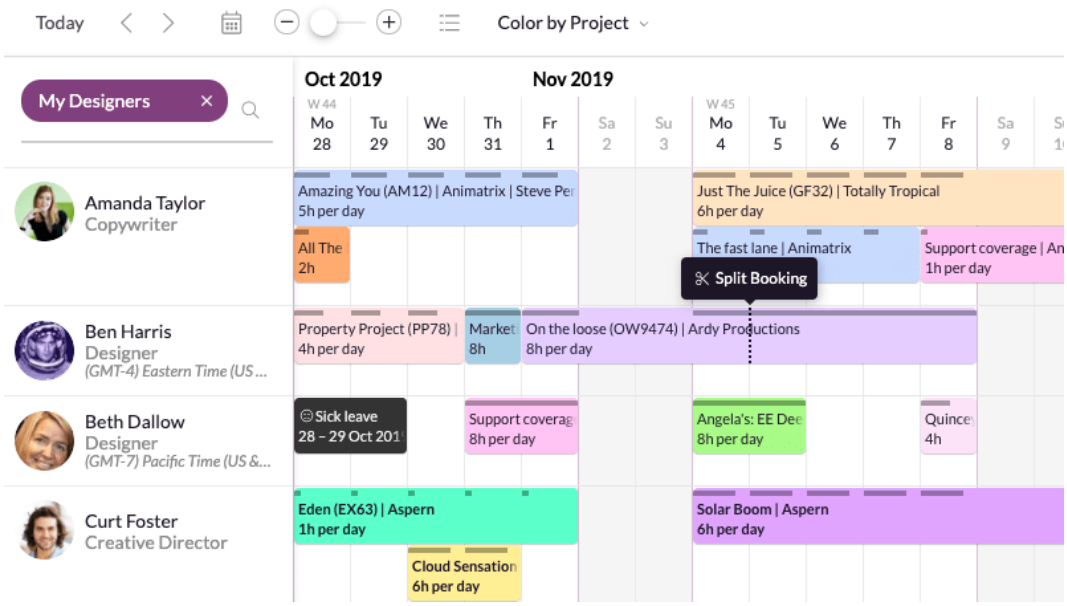

Figure 10: Resource Guru - Overview

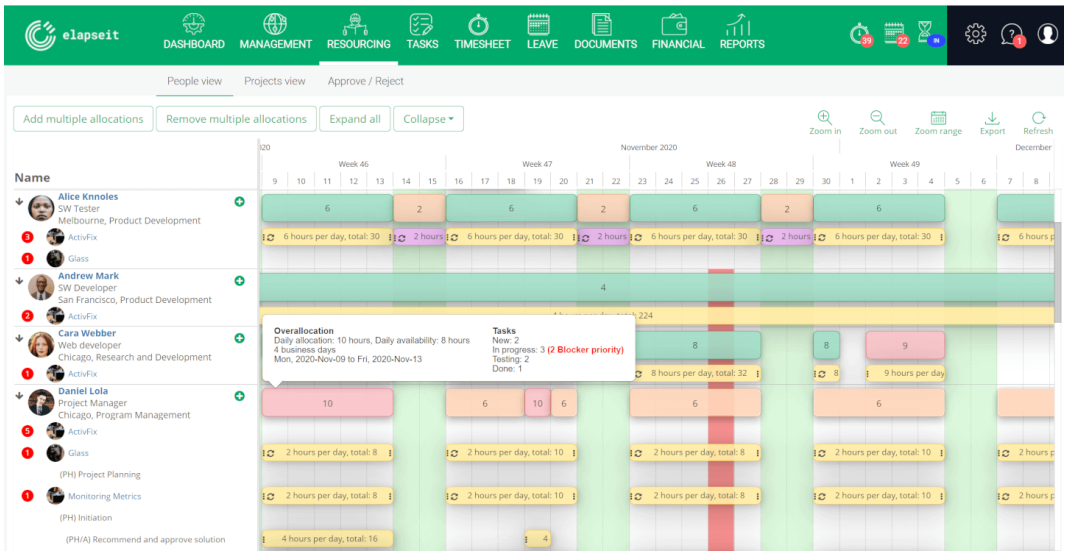

Figure 11: Elapseit - Overview

3.4 DISCUSSION

After experimenting with the solutions, they were ranked using the previously selected criteria (see Table 1). Each criteria was ranked individually for each software with a value on the following scale[6]:

1. (Fail) Unacceptable. Poor. Much less than acceptable.
2. (Fail) Weak. Less than Acceptable.
3. (Pass) Good. Acceptable. Satisfactory. Average.
4. (Pass) Very Good. Full Performance Behaviours. Above average.
5. (Pass) Excellent. Exceptional. Mastery. Much more than acceptable.

After analysing the scores of each program, the reached decision was that Runn will be the main inspiration to future solutions, considering it had a greater score in most of the categories when compared to all the other choices. Besides this decision, we also concluded that Mavenlink's data analysis reports can be used as inspiration for possible features, and that the other choices can also be used as inspiration for useful interactions.

6 Scale used for ratings: https://www2.gov.bc.ca/assets/gov/careers/for-hiring-managers/resources-for-hiring-managers/5_point_rating_scale.docx

Criteria	User Interface (UI)	Usability	Workload Resource Planning	Interactions	Timesheet Integration	Automation	Alerts
Runn	5	5	4	4	5	4	5
Mavenlink	4	3	3	3	5	4	4
Float	4	4	3	4	4	4	4
Resource Guru	3	2	2	3	2	4	2
Elapseit	2	3	5	4	5	4	4

Table 1: Software Criteria Ranked from 1 to 5

4

CREATING A SOLUTION

It was mentioned in Section 2.2.1 that there are two types of prototypes that can be created: low-fidelity or high-fidelity. It was also mentioned in Section 2.3 that there are steps that can be followed in order to achieve a good final prototype. In the suggested steps, the old design should be tested, to identify the good and the bad parts; then, user research should be done to understand how users behave; then, the first prototypes can be sketched in paper; after that, the best ideas from the low-fidelity prototype can be included in a new high-fidelity prototype; this prototype can then be tested until no problems are encountered. However, considering that the target audience has reduced availability for user testing, the steps were slightly rearranged in order to occupy less working time. In this chapter, the solution creation process will be described and explained, using some of the research done in Chapter 2 to achieve the best version possible.

4.1 FIRST ITERATION OF PROTOTYPES

The first step in creating the design was sketching a low-fidelity prototype that could be showed later to a group of managers in order to gather feedback.

As mentioned in Section 2.2.1, the designs can follow two different directions: convergence or divergence. Because this will be an internal tool, and not a tool to be marketed, the convergent approach was chosen, i.e. joining two or more designs toward a central solution.

Another topic considered was the choice between a design that emphasizes recall (e.g. command line commands) or recognition (e.g. buttons or menus), both mentioned in Section 2.1.1. In that same section, three suggestions were also given to help reduce the gap between the demands placed on the user and the user's capabilities: considering the frequency of use, if the use will be mandatory or discretionary, and the knowledge level of the user. When choosing what option would be the best, we had to consider what kind of user will use the solution. The majority of VILT's employees have a background in STEM (Science, technology, engineering, and mathematics) areas, so it is safe to say that most people would, for example, understand a command line approach to create a new allocation, and it would be a more effective way to interact with the solution. However, this would increase the time it takes to learn how to use the tool, which can help decrease the adherence from the users, and that is not the objective. The objective is creating a solution that is intuitive and easy to learn and use, so that even though the solution will not be of mandatory use, the users still want to use it. For this reason, the recognition approach was chosen for the majority of the design, making the search bar an exception that

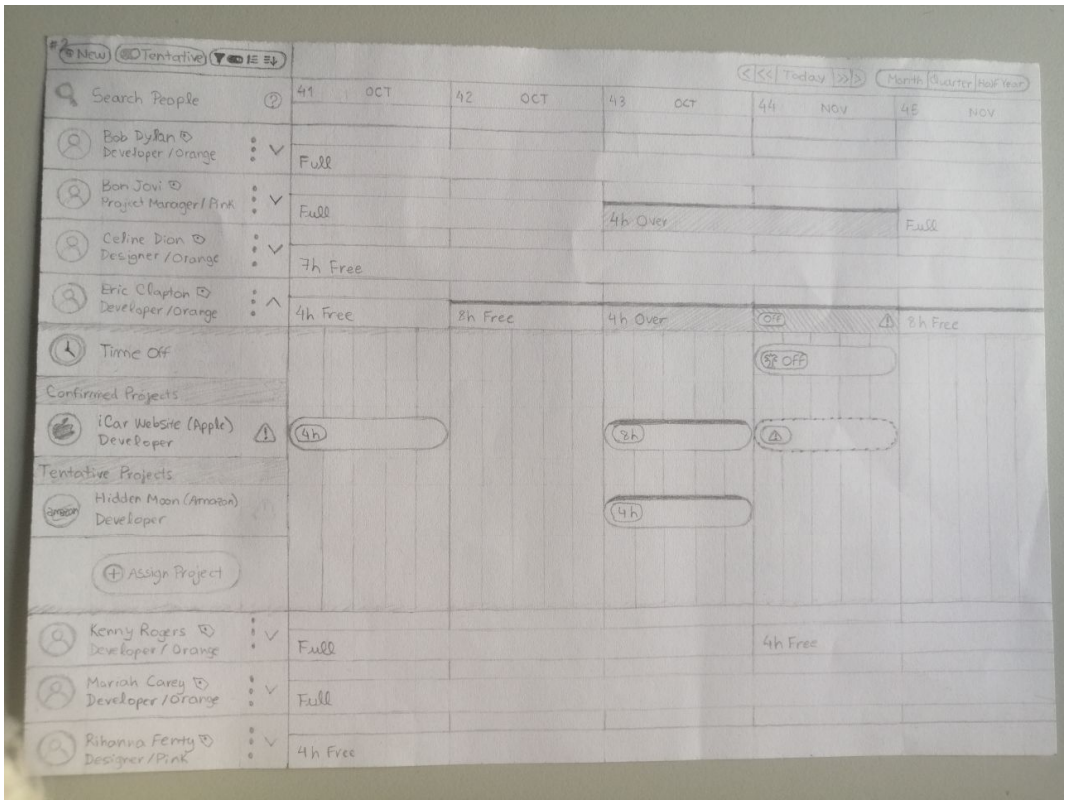

Figure 12: Low-Fidelity Prototype - Design

combines both recall and recognition: the user using the search bar will need to recall the words by memory, but as soon as one is written, a list of matches will appear, allowing the user to visually recognize the desired choice.

This first mock-up was sketched with pencil on paper (see Figure 12), and was mainly inspired by Runn's design, as decided in Section 3.3.

In order to increase the amount of feedback collected when studying the target audience, an alternative version of the first mock-up also created, but defining possible interactions and functionalities that could be added (see Figure 13). Unlike the visual design, these were inspired by the software choices shown in Section 3.3, and not just Runn. The chosen interactions (and software that inspired them) were:

- Click on an empty block to create allocation (Runn)
- Click + drag on an empty block to create longer allocation (Runn)
- Hovering over an allocation will suggest splitting it (Resource Guru, Runn)
- Hold + drag an allocation to reallocate it to another date (All)
- Press Shift + drag an allocation to duplicate it (Float)
- Drag ends of an allocation to expand/shorten it (Float)

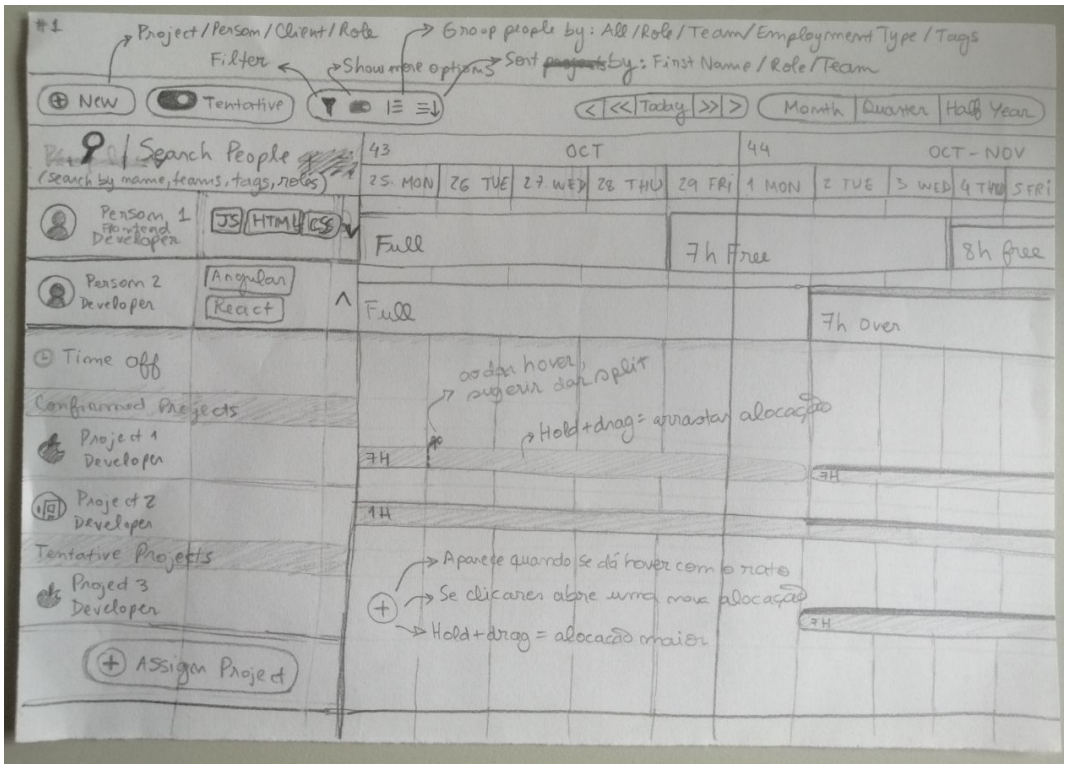

Figure 13: Low-Fidelity Prototype - Interactions

These will be discussed later with the managers, to see which features are considered useful or not.

The main goal for this step was creating an initial idea that could help the managers by gathering as much feedback as possible in a short amount of time.

4.2 PROBLEM RESEARCH

To better understand the problem, a survey was then realized and answered by VILT's Professional Services managers. This survey was conducted through a form, created using *Google Forms*, and sent to the company inbox of 15 managers. Of those 15 managers, 6 managers answered the form. This form can be consulted in Annex A. The form was divided in 4 sections:

1. Introduction
2. Using Zeus' Allocation Map
3. Using Other Methods
4. Interview

The answers were gathered and studied, to create a better prototype that takes users' needs into consideration.

33.3% of the users use the current allocation map, whereas 66.7% of the users use alternative methods. Gathering the answers of the users who *use the current Zeus allocation map* (2 users), these were the collected answers:

When asked about why they still used the current allocation map:

- They use Zeus allocation map because they consider it useful, considering it is the centralized information system and has a lot of the information.
- They also use an excel besides that to manage everything better.

When asked about what they like about the current allocation map:

- It is useful to have information in a calendar view.

When asked about their current routines for managing allocations, the different approaches were:

- Select users one by one and check allocations.
- Use Excel spreadsheet to see the timeline, use Zeus allocation map as the point where to store the information in the end, and use *Metabase* Dashboard to see the future expected allocations.

When asked about what improvements they wished to see:

- Filters that allow seeing the whole department allocation.
- Show when people are on vacation.
- Bigger variety of filters.
- Improve usability in order to add new allocations.
- Having a useful view of the full map with the manager's team.

Gathering the answers of the users who *use alternative methods* (4 users), the collected answers were:

When asked about which methods they use as a replacement, there were answers that included more than one method. The mentioned methods were the following:

- Excel spreadsheet (4 users)
- Metabase[1] (this is, a software feature in Zeus that helps visualizing analytics and implementing personalized database queries) (1 user)
- Paper annotations (2 users)

1 https://www.metabase.com

- Local database (1 user)

When asked about their current routines for managing allocations, the different approaches were:

- Go to the excel spreadsheet, find the people, update the allocations.
- Every Monday, do a check of the allocations for the week, in order to see if there are any holes. Whenever a new allocation comes along, update the spreadsheet (with a '?' if it's not confirmed).
- Make a weekly review for the entire allocations map. After that, update manually the excel sheet that serves as a base to feed a local database. From this database, it's possible to extract relevant and real-time data for allocations. Some of this data is analysed using *Metabase*.
- Have a team chat and when a new task arrives, the person with less work starts working in the new task/incident.

When asked about the advantages of the alternative method compared to the current allocation map:

- See the whole department in one sight.
- Easy to move and erase allocations.
- Easy to copy and paste one week to the other.
- Quick to add things in a visual way.
- Flexible interface to manipulate data.
- Better *UI* and usability.
- Relevant views that allow having a better perception of the allocation of a team or person.

When asked about what features they would like to see in the improved version of the allocation map:

- Alarms (when an allocation is about to end, when there are conflicts, etc).
- Easy way to get an overview of the team (Show all resources on a single page).
- Easy way to prolong an allocation.
- Allow view/edit/manipulate all data in the same screen.
- Improve UI.
- Upgrade the calendar view.
- Create filters by skills, in order to search resources in the future.
- Ability to extract different maps filtered by customer, period range, resource title and others.

The last question asked the participants if they would be willing to be interviewed. Of the 6 users who answered the form, 5 of them answered "Yes" to this question.

This could be seen as a low number of interviews. However, it was not considered harmful to this study, considering more iterations of the initial prototypes will be done, and five participants is enough for a good test [9].

After the answered forms were received, and there was a confirmation that no more submissions would happen, the 5 managers that chose "Yes" to the last question were interviewed via video call. For this, a slideshow presentation[2] was created with the screenshots of the existing software shown previously in Section 3.3, as well as the mock-up that was initially sketched. In each interview, the same procedure was executed. First, the participants were shown *Runn* being executed. The participants were asked to be as open as they could about their opinions towards the software, so that more feedback could be collected. Then, the participants were shown the presentation and asked what they liked and disliked about each one, as well as other useful ideas they might have. In the end of the presentation, the participants were asked a specific set of questions related to components and options that appeared differently in the presented software choices. The questions were:

- Which set of 3 time ranges would be the most useful? The options are: 15 days, month, quarter (4 months), and half year (6 months).
- Which representation of the allocated time is the most useful? The options are: Daily hours or percentage.
- Do you think it is useful to show weekends in the allocation map?
- Are there design preferences that you would like to see in the solution?
- Are there interactions or functionalities that you would like to see in the solution?

These were the insights and ideas gathered during the whole process of all interviews:

There was divergence of opinions when asking about the preferences in time range of views. The majority chose longer time ranges like month, quarter, and half year, while a minority chose shorter time ranges like 15 days, month, half year.

Although most people stated that showing weekends is mostly unnecessary, there were people who requested that at least the shorter time range view should show weekends, and at least present a clear distinction between different weeks.

It was also noted that Zeus keeps track of every vacation, time-off, vacation request, and other kinds of work time disturbances, so these should be automatically updated in the allocation map. Besides time-off, Zeus also keeps track of every client and project, so the allocation map page will not need to create new ones, only import them.

When asked to choose between daily hours or percentage to split multiple projects in the same time allocation, everyone chose percentage, because not every employee works the same amount of time and it makes it easier to manage it.

2 https://www.beautiful.ai/-MmIj4XZ_OetaCXN-V37/

When asked about the design, the suggestions were simple colors, and a clear distinction between different elements.

Important suggestions for features and interactions were: dragging the ends of a block to increase/decrease it, optional notes and tags for each employee, the possibility to give each employee a role in a project, showing availability by role in a given time range, having a checkbox to decide if weekends and holidays should be included in the created allocation, the possibility of creating a personalized search with multiple selected people, and including a variety of filters to choose from such as technology, knowledge, idioms, certificates, etc.

A few additional features that were suggested "if there was extra time" were: possibility of reallocating a time allocation to another person, not allowing an allocation to be created after someone leaves the company, etc.

4.3 REQUIRED FUNCTIONALITIES

In order to define the best interface, we used the information gathered in the previous section to define a list of functionalities that the solution should include. The list was initially created using the old allocation map as reference and, after that, improvements were added based on the answers received in the previous section.

First, the initial screen was planned out. When entering the page, a user with manager permissions will be able to:

- Add to search or clear search
- Modify view
- Refine view with filters
- Create or modify an allocation
- Request a report

After the initial screen's main functionalities were defined, a more detailed list was created for each item.

The user will be able to create a personalized search that better fits their needs, by the use of the planned functionalities:

- Add a person to the search
- Add a team to the search
- Add everyone to the search
- Clear search

The user will be able to modify the view in terms of time, i.e., the user can adjust the time range in order to show more or less allocations per user:

- Switch to view by month

- Switch to view by quarter
- Switch to view by half year
- Go to the left/right by a short period of time (< | >)
- Go to left/right by a long period of time (« | »)
- Go back to today's date

Besides the personalized search, the user can also apply a range of filters, not related to a user or team's name, but to their availability or other desired fields:

- Select filters
- Reset filters
- Apply filters

Each user shown in the allocation map may or may not have assigned allocations in each time slot. Creation, edition, and deletion of allocations are essential functionalities:

- Create new project allocation
- Create new opportunity allocation
- Create new undefined allocation
- Change allocation details
- Reassign allocation
- Delete allocation

Lastly, the possibility of requesting a report file including specific details shown in the allocation map was added:

- Select report details
- Export report

4.4 SECOND ITERATION OF PROTOTYPES

In order to achieve the desired solution, the objective is to create the most realistic and interactive prototype possible, so that the managers can analyse and evaluate the solution, and so that future developers can implement the solution accurately according to this study.

Before starting the new prototypes, we first looked at the old allocation map (see Figure 2), and then took into consideration the feedback gathered from the interviews with the managers. Besides wanting to create a solution

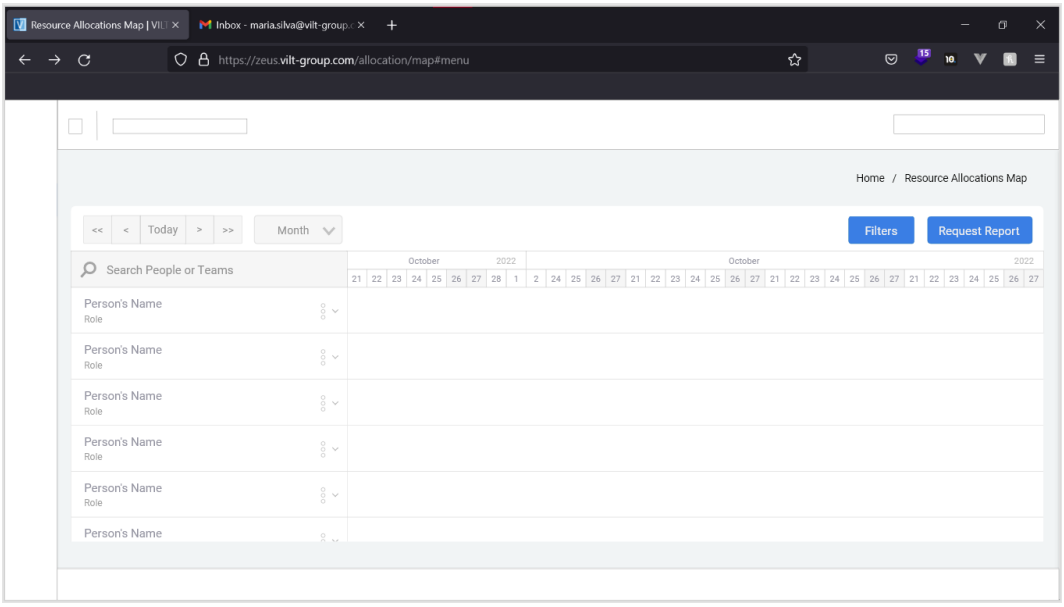

Figure 14: Base for the different month views

with a bigger range of functionalities, we also want to create a more visually appealing solution, which will make the managers want to use and stick with the solution in the end.

As mentioned before, different sets of prototypes should be designed and tested before the final version is decided. In this section, the first iteration of high-fidelity prototypes that were created and later showed to the managers will be presented. Three sets of prototypes were created for three different time ranges: a month view, a quarter view, and a half-year view. These prototypes were created using the software *Adobe XD*.

4.4.1 *Month View*

We started by designing a realistic prototype of what the solution would look like. We wanted to have different designs that could be presented to the managers, so the base layout was designed first (see Figure 14).

On the top left side of the screen there are two components to change time. The first component is a set of 5 buttons, that can make the calendar go back or forward in time, or show the current date. The second component is a drop-down menu that allows the user to change the view to the different time ranges: Month, Quarter, and Half Year. The default time range is the Month view. On the top right side of the screen there are two buttons: Filters, to select different filters and Request Report, to allow the user to download a report with the information that is represented in the allocation map.

Under these components, there is the actual allocation map. There is a list of employees that are presented by default. Each row represents an employee. On the left side of each row is the information about the employee, as well as an options button. There is a search bar above the list of users. The search bar allows the manager to browse the existing users and select a new user to add to the list that is presented in the allocation map.

The dialog style guidelines mentioned in Section 2.2.2 were considered, and the two types of chosen dialog styles were: menus (for the choice of the time range) and direct manipulation of objects (for the buttons and direct interactions with the allocation map, e.g. reallocating an allocation to a different date). For the menus, the three possible choices (Month, Quarter, Half Year) are consistent, mutually exclusive, and ordered in ascending order of time range. For the direct manipulation of objects, a separation of the components was done purposely to help the user, reducing the semantic distance. This way, the user can associate the left side of the top bar to time changes, and the right side of the top bar to settings and information about the allocation map.

Inspired by *Runn* as seen in Section 3.3, each person's row in the solution will contain two inner fixed rows: Time Off, to represent confirmed vacations, pending vacation requests, medical leave or holidays, and Learning, to represent the time allocated to learning and training. Other than these two fixed rows, more rows can be created to assign projects or opportunities to a person.

Because the software should be prepared for imperfect use, we also created responses to potential error situations. These error situations include over allocation, i.e. when more time is allocated than what is possible; or when an allocation is created during time off.

A specific color scheme is proposed to represent:

- Time Off - Gray
- Learning/Training - Orange
- Projects/etc - Blue
- Error - Red

This color scheme was chosen for two types of reasons: the company's colors, and color theory. To create the color scheme, we wanted to include VILT's main colors: blue and orange. These are two complementary colors in the color wheel that are associated with the VILT brand, so these colors are already present in Zeus. The color blue was chosen to represent regular allocations, since this color is associated with information and tranquility [11]. The color orange was chosen to represent the Learning rows, since it is a vivid color that is associated with creativity [12]. Besides these two colors, the color grey was chosen to represent the Time Off rows. Because this color does not have saturation, it can be associated to the feeling of being away, out of work, inactive[3]. The color red was chosen to represent the error situation, since it is commonly associated with alerts, errors, and danger [13].

During this creation process, the usability guidelines mentioned in Section 2.2.2 were also considered. The language used is consistent and familiar to the user (no new terms were added, just the ones discussed with the managers). Color coding is used uniformly, as shown in the decided color scheme. The solution allows user to understand if there are current errors in its use (by the use of the color red). In order to make the solution more efficient, specific interactions were planned to make the solution easy, fast, and efficient to use (e.g. dragging an allocation to change the dates instead of editing the dates one by one). To help the user with less need for storing memory, "see and point" methods were used more frequently than "remember and type" methods, by

3 https://www.sensationalcolor.com/meaning-of-gray/, last visited 24/10/2022.

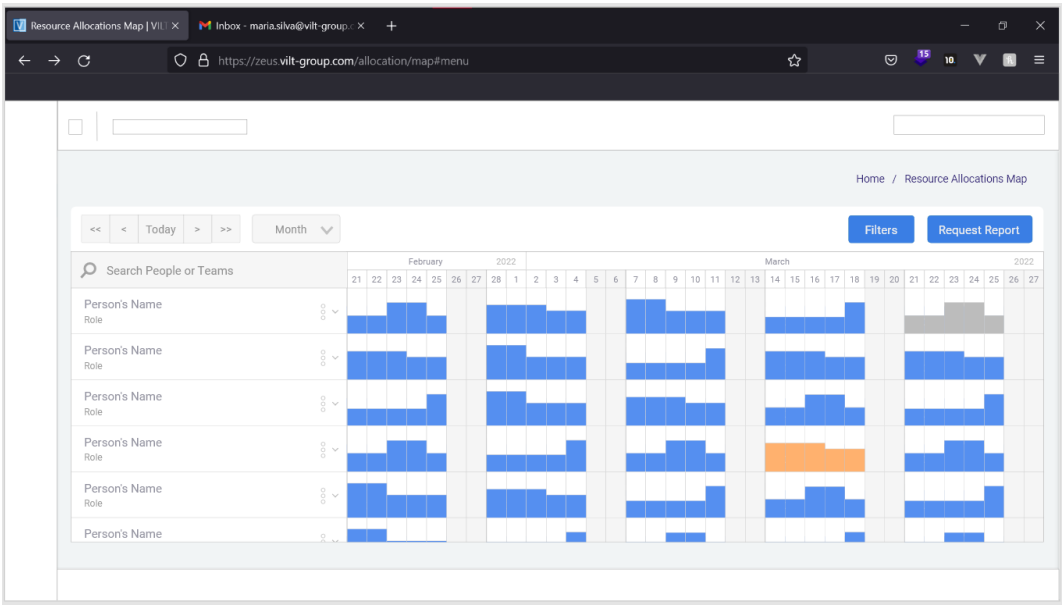

Figure 15: Month view - Bars

using buttons and lists of choices. The created design is simple and consistent, making the user less prone to becoming overwhelmed by the complexity of colors or amount of elements.

Four different designs were created initially. The difference between each of these is the way the allocated time percentage is visually represented, i.e. how we show how much time is occupied or not by each user. These designs did this by showing changes in: bar size, percentage, opacity, or fixed opacity (the differences between these will be explained later in the text).

In the first design (see Figure 15), bar size changes are used to represent how full is the allocation. If an employee has their time allocated to 100% in one of the days, it should appear as a full sized bar. If less than 100% is allocated, a smaller bar should appear in that day. This way, the managers can distinguish easily when each person is free or busy without needing to open the person's row and without the need to read any information.

The second design (see Figure 16) used a percentage to represent how full an allocation is. This one feels harder to interpret all at once, considering the manager needs to read the percentages one by one. But it can be more accurate when reading how much time of each day is occupied.

The third design (see Figure 17) makes each allocation's opacity the same as the allocation percentage. This is an easy way for the managers to quickly interpret the data, but makes it difficult to distinguish if there are slight changes in opacity.

The fourth design (see Figure 18) is very similar to the previous one, except that there are slots with fixed opacity, instead of using the same opacity as the allocation's percentage. If an allocation is 100% full, it will have 100% color opacity. If it's less than 100% full, it will have 70% color opacity.

After designing the initial visual representation of the solution, we started by designing essential interactions. It is possible to scroll through the list, and when clicking on the person's row, this component expands and allows the manager to visualize the existing allocations (see Figure 19). When hovering over a free day, the manager

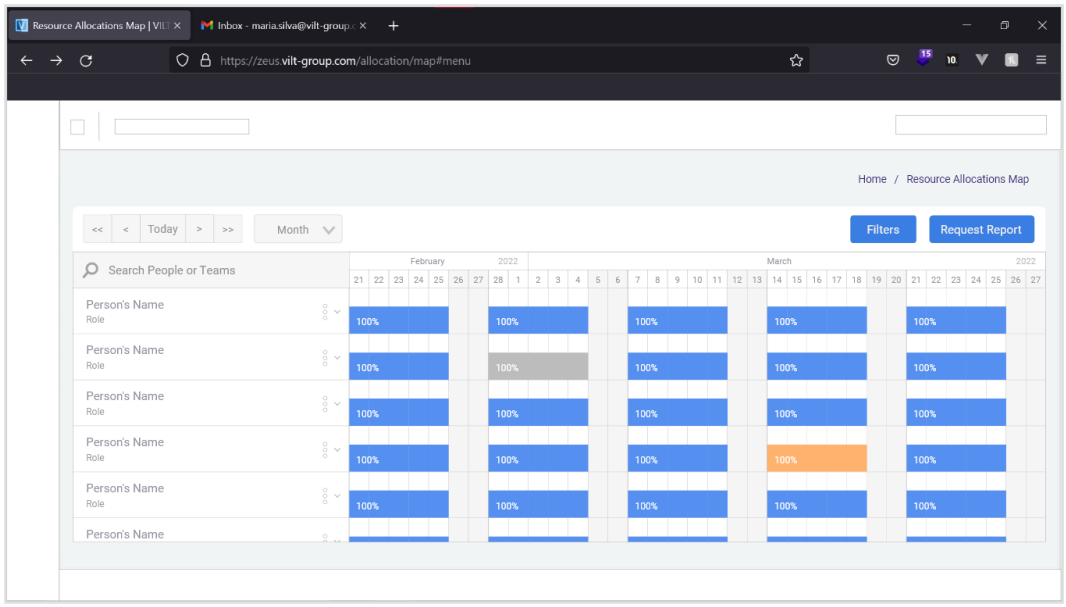

Figure 16: Month view - Percentage

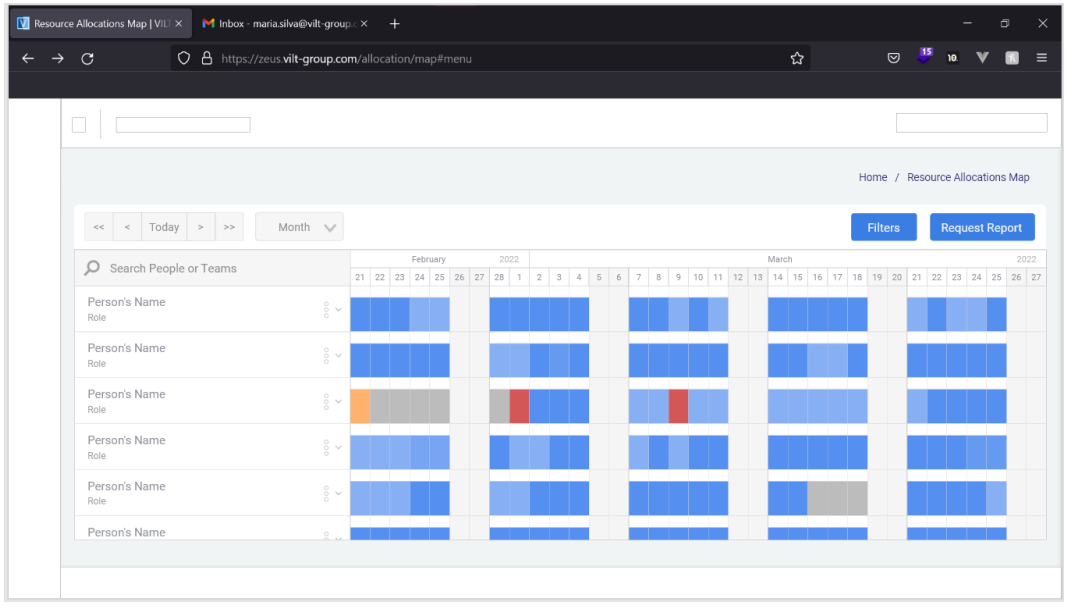

Figure 17: Month view - Opacity

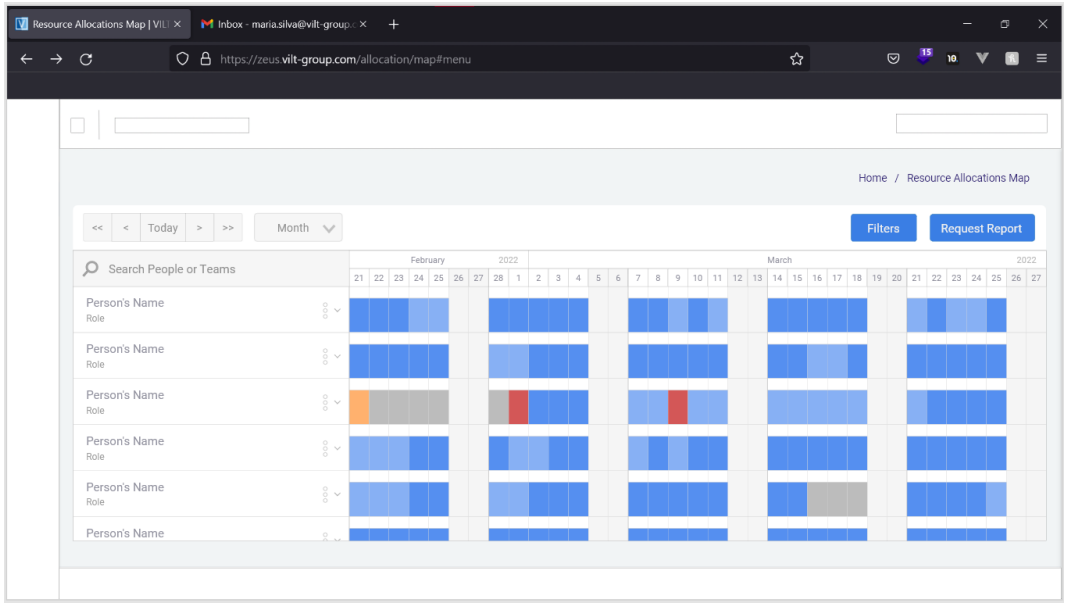

Figure 18: Month view - Fixed Opacity

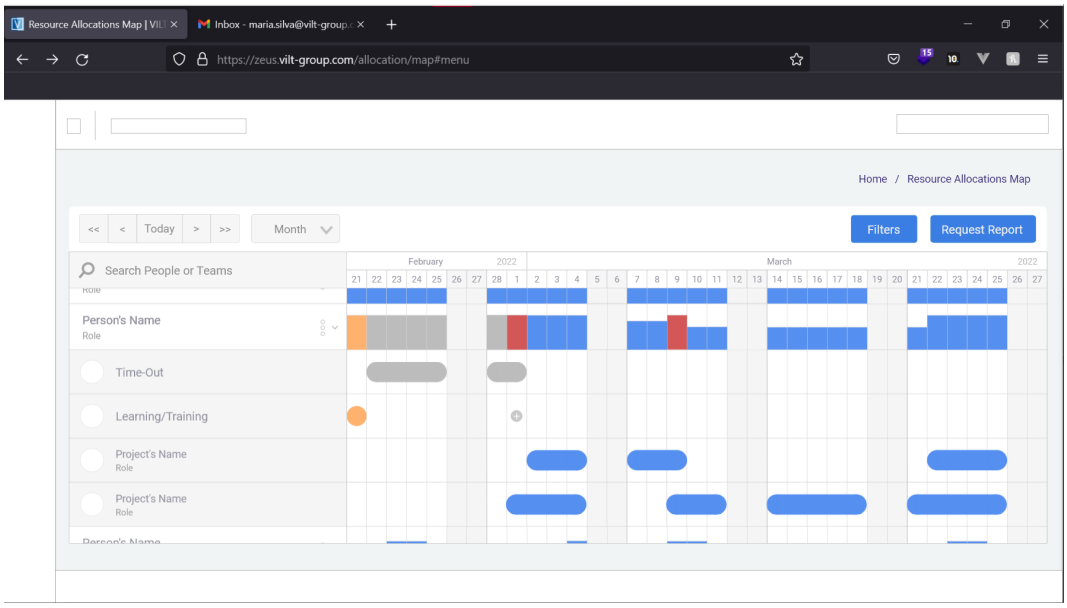

Figure 19: Month view - Open Row

sees a button to create an allocation, which can be clicked (to open a pop-up) or dragged (to directly create the allocation). When clicking on an existing allocation, the manager can modify it or delete it. When a project is not allocated to a specific user yet, the manager can do it by clicking the Assign Project button, or dragging one of the day columns. This will open a pop-up to select which project is to be selected.

In order to create the best solution possible, before creating the final prototypes, we want to have answers to some questions that we thought would be best answered by the managers, since they have experience with these tasks:

1. *Should there be separation of categories or people by colors?* Possibilities include adding a color attribute to each person depending on their role, or using each person's favorite color (which already exists in the table "user_details" of the database).

2. *Is it necessary to include a divisor between projects, opportunities, etc, or is it enough to have a specific icon for each category?* When opening a row corresponding to a user, multiple rows will show up. We can either separate projects and opportunities with a fixed separator row, separate the project category by color or icon, or choose no separation at all.

3. *What is the preferred way to indicate a full time slot?* The possibilities are representing the time percentage through opacity changes, bar size changes, or even both. In either case, it seems important that there's a white space between people, or a guideline to make it visibly clear when an allocation is full or not.

4.4.2 *Quarter View*

We created visually similar views to allow the visualization of 3 months. For this time range, we didn't create the four different designs, but instead we chose what at seemed like the two best candidates: bar size changes (see Figure 20), and fixed opacity changes (see Figure 21).

We also created the prototype of a row being opened to show the allocations in more detail in the quarter view mode (see Figure 22).

4.5 FINAL PROTOTYPES

After presenting the first iteration of prototypes to the managers, we asked the two supervisor managers who are associated with this project to answer the questions mentioned previously. These answers were considered in the creation of the final prototypes.

1. *Should there be separation of categories or people by colors?* No. It was considered unnecessary, so there will be no differences of color in each person's row.

2. *Is it necessary to include a divisor between projects, opportunities, etc, or is it enough to have a specific icon for each category?* It was decided that separating the project category with different icons would be the preferred approach.

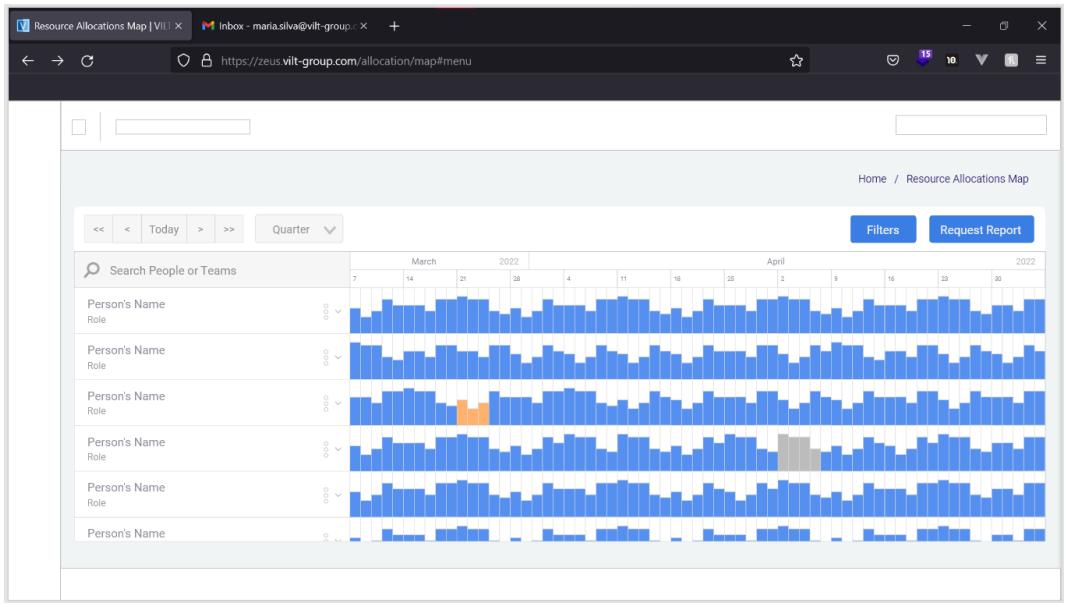

Figure 20: Quarter view - Bars

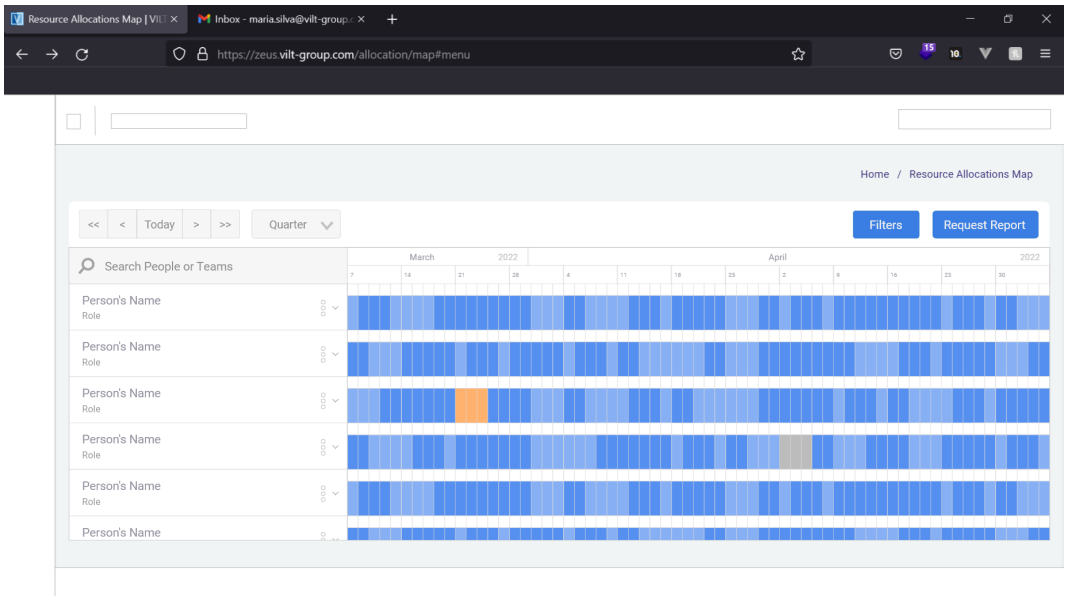

Figure 21: Quarter view - Fixed Opacity

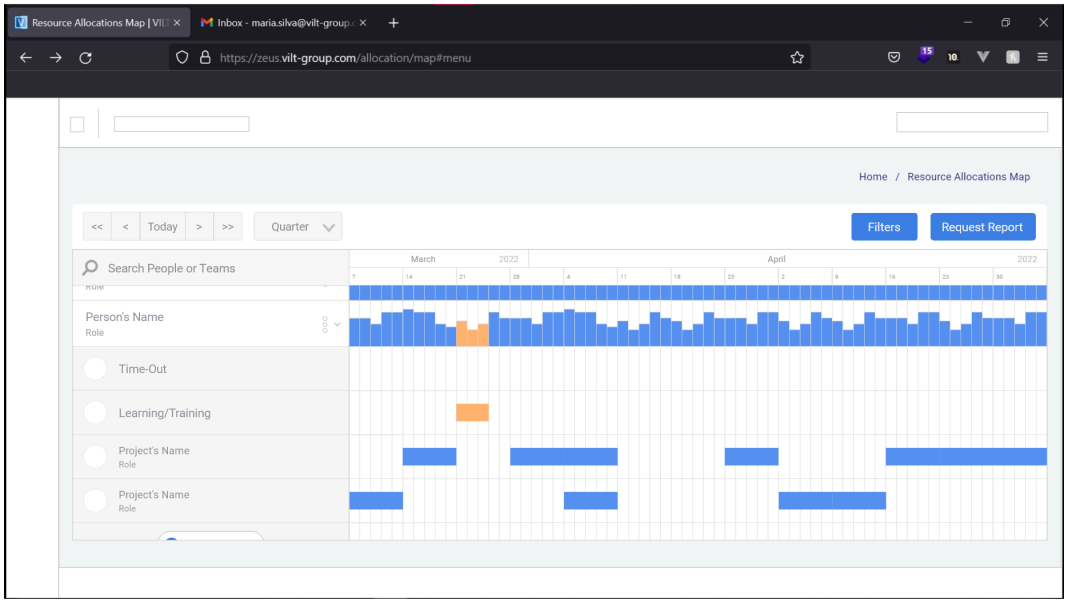

Figure 22: Quarter view - Open Row

3. *What is the preferred way to indicate a full time slot?* The conclusion was that the easiest method to visually interpret the time allocations would be a mix of two designs: fixed opacity and bar size changes.

These conclusions led to the final prototypes, which were also created using the software *Adobe XD*.

Month View

First, we have the month view prototype (see Figures 23 and 24). This will be the first screen the user will see when entering the allocation map page.

Besides the answers collected to the questions about the previous prototypes, the managers had one more request: the creation of a compact view, for viewing more rows in the same amount of space, and a way to switch between views. For this, the switch button was created, as well as the Compact Mode prototypes (see Figures 25 and 26). These were created for each time view (month, quarter, year). This view makes the rows smaller in height, and adds a button in every prototype to allow switching the views between normal mode and compact mode.

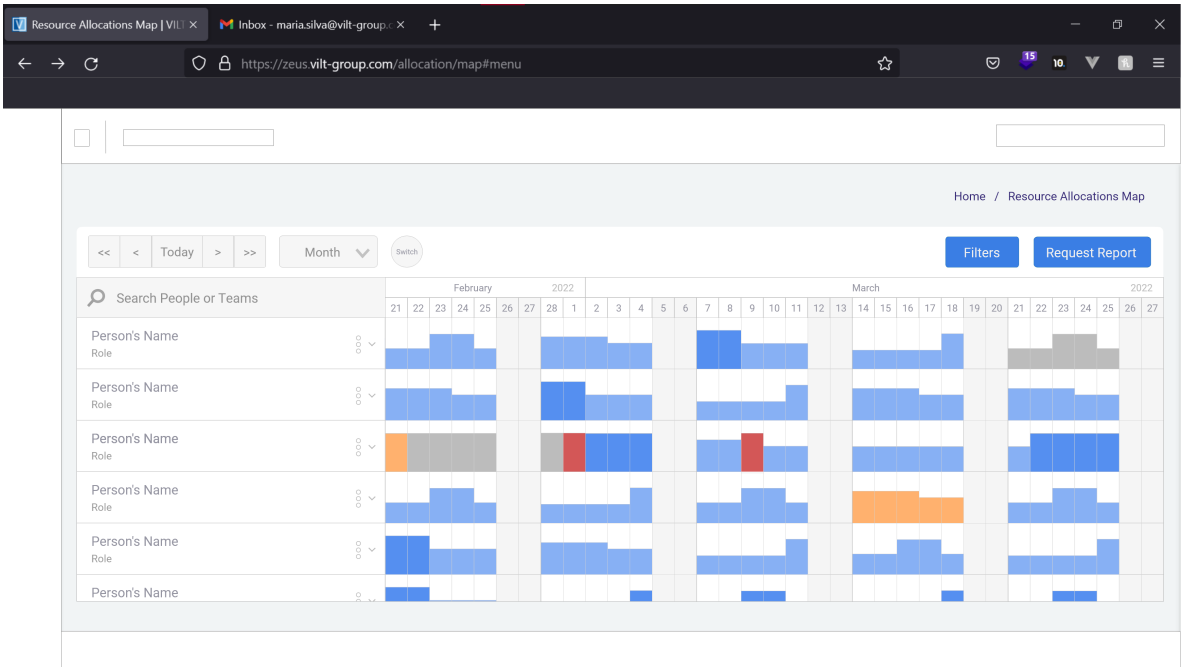

Figure 23: Month View

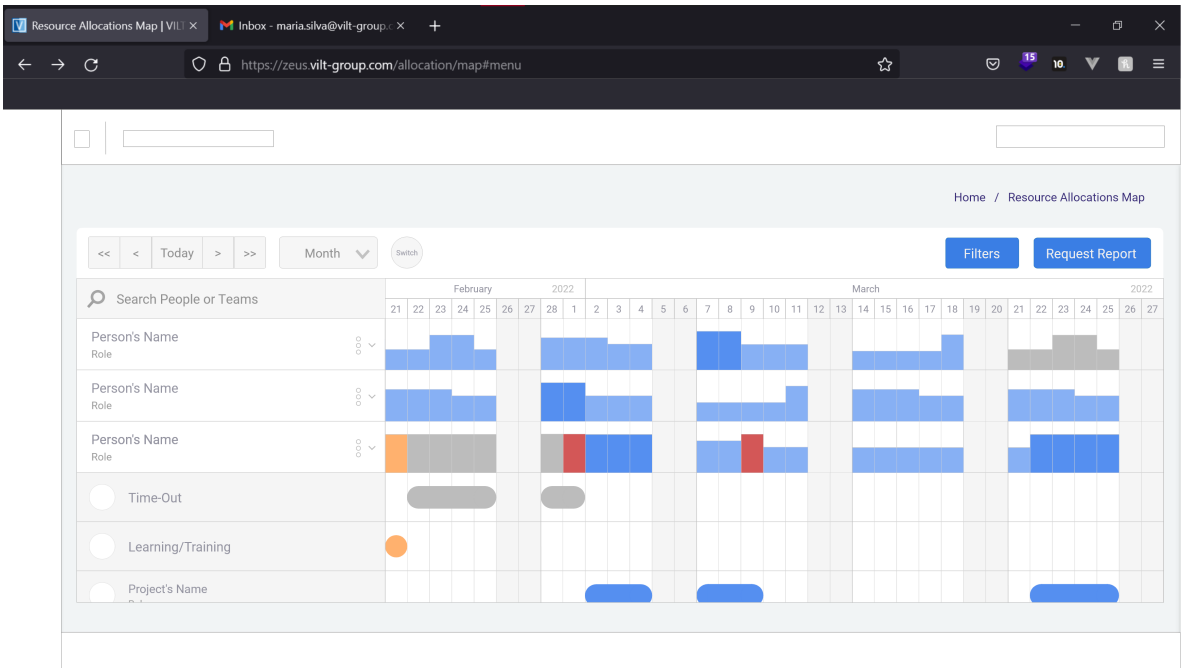

Figure 24: Month View - Open Row

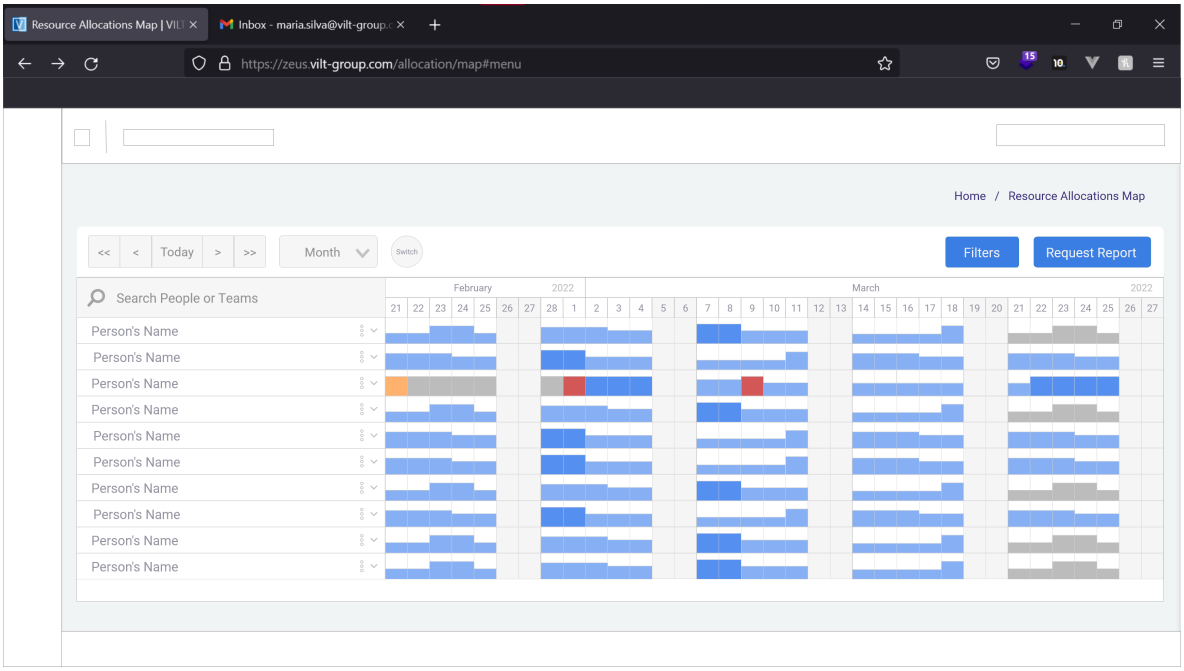

Figure 25: Month View - Compact

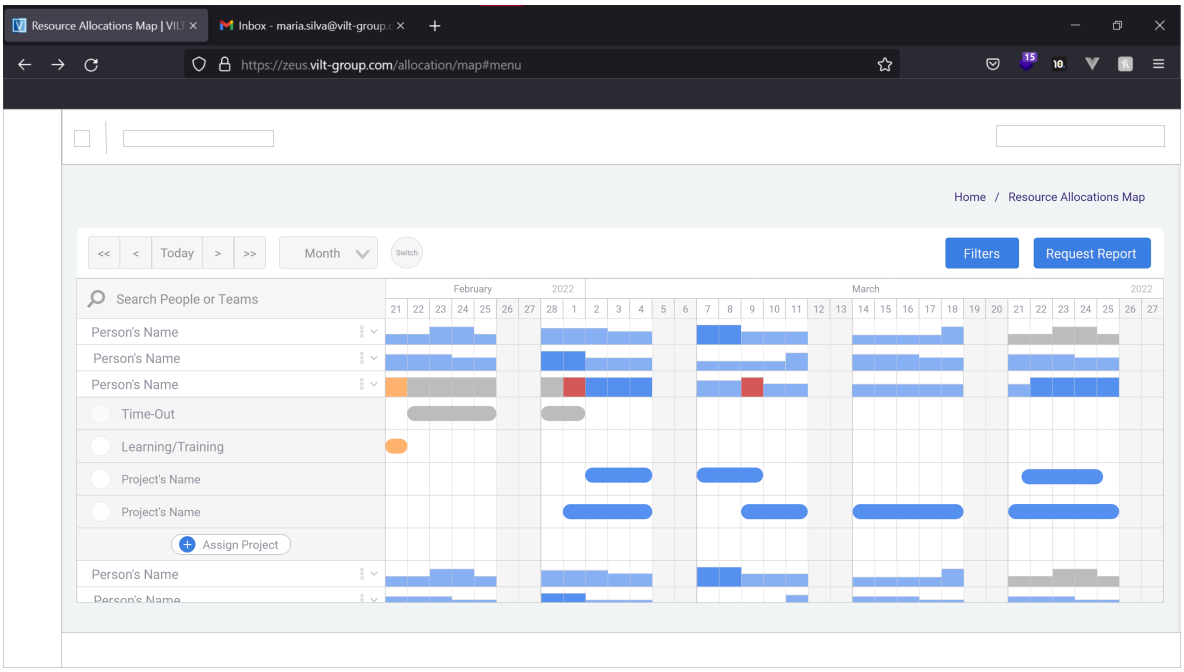

Figure 26: Month View - Compact - Open Row

Quarter View

After that, the quarter view (see Figures 27 and 28) was created, following the same design as the previous one, as well as the Compact Mode views (see Figures 29 and 30).

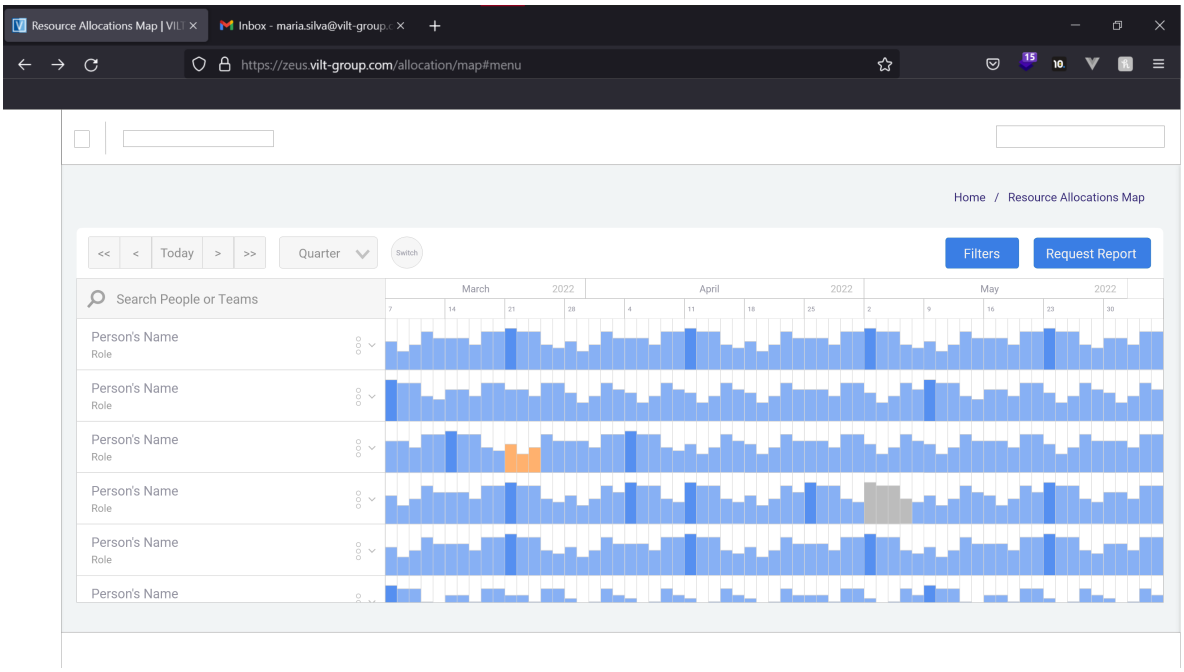

Figure 27: Quarter View

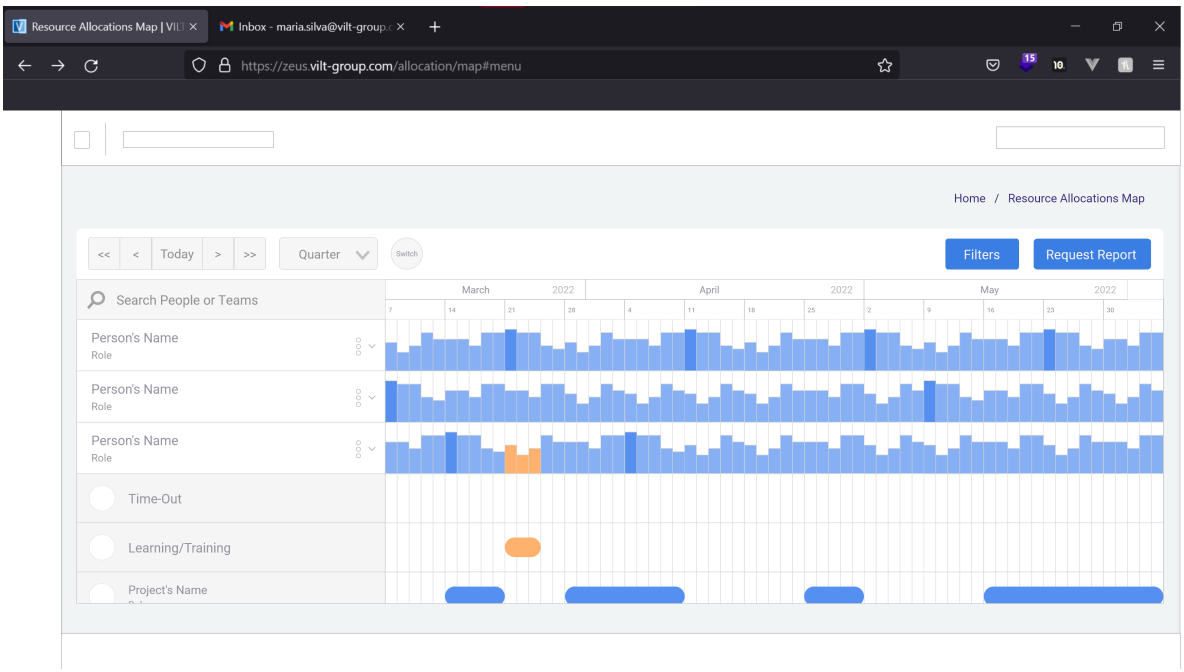

Figure 28: Quarter View - Open Row

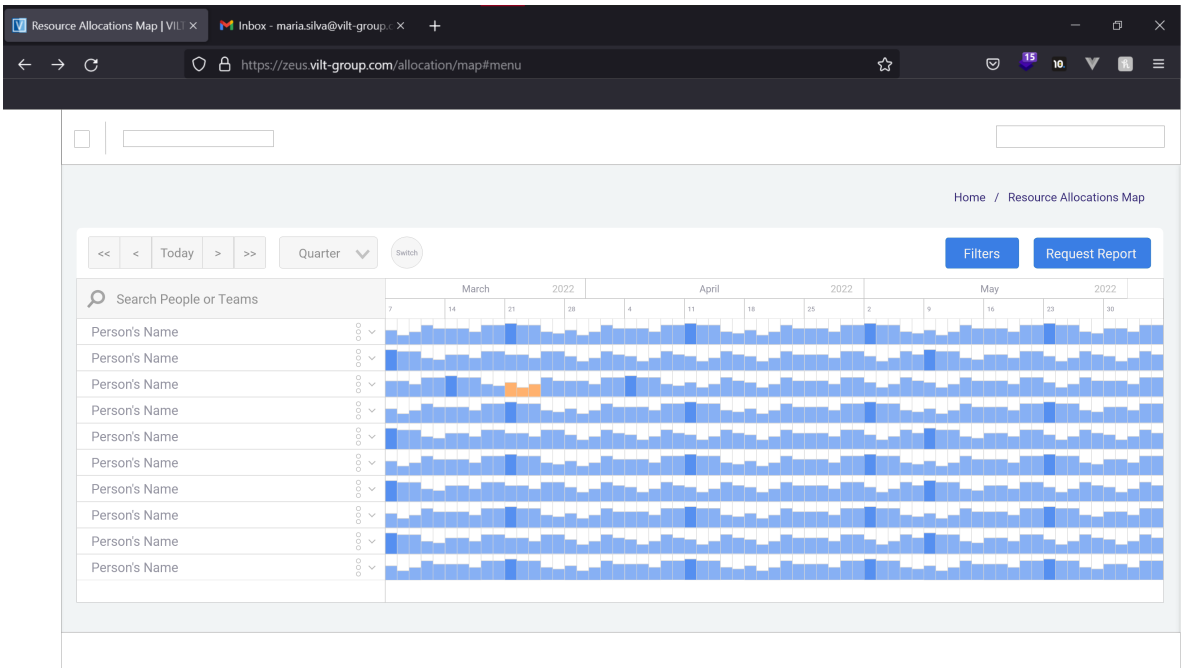

Figure 29: Quarter View - Compact

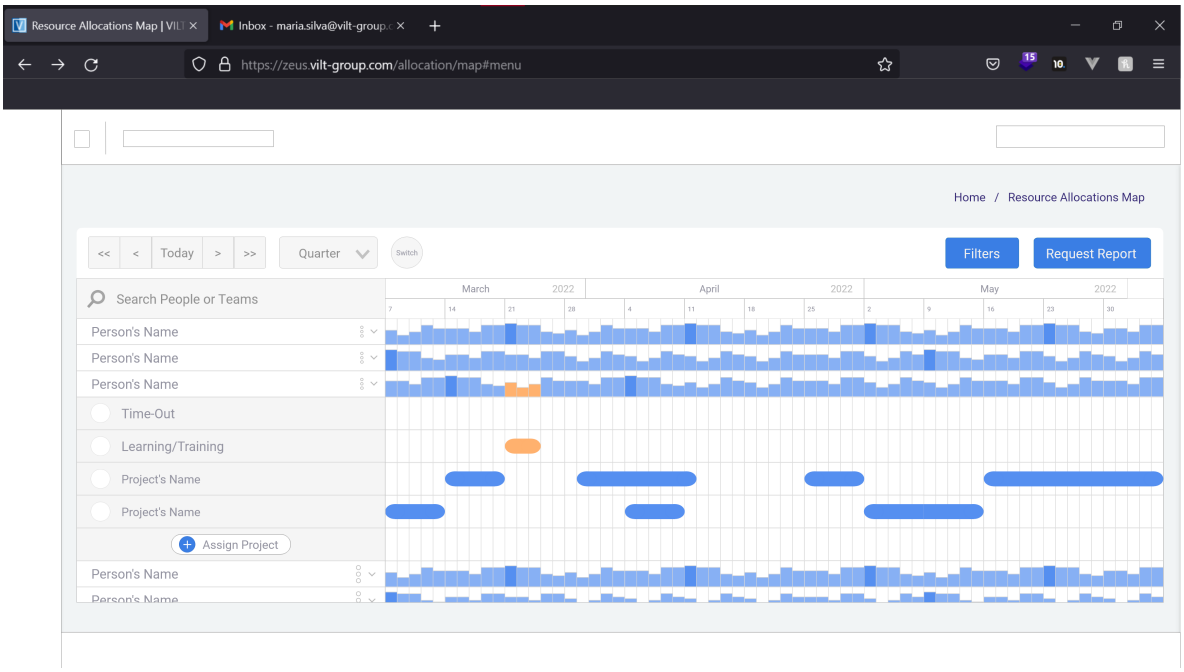

Figure 30: Quarter View - Compact - Open Row

Half Year View

And finally, the half view (see Figures 31 and 32) was created, allowing a more generic and long term view of longer projects, as well as its Compact Mode version (see Figures 33 and 34).

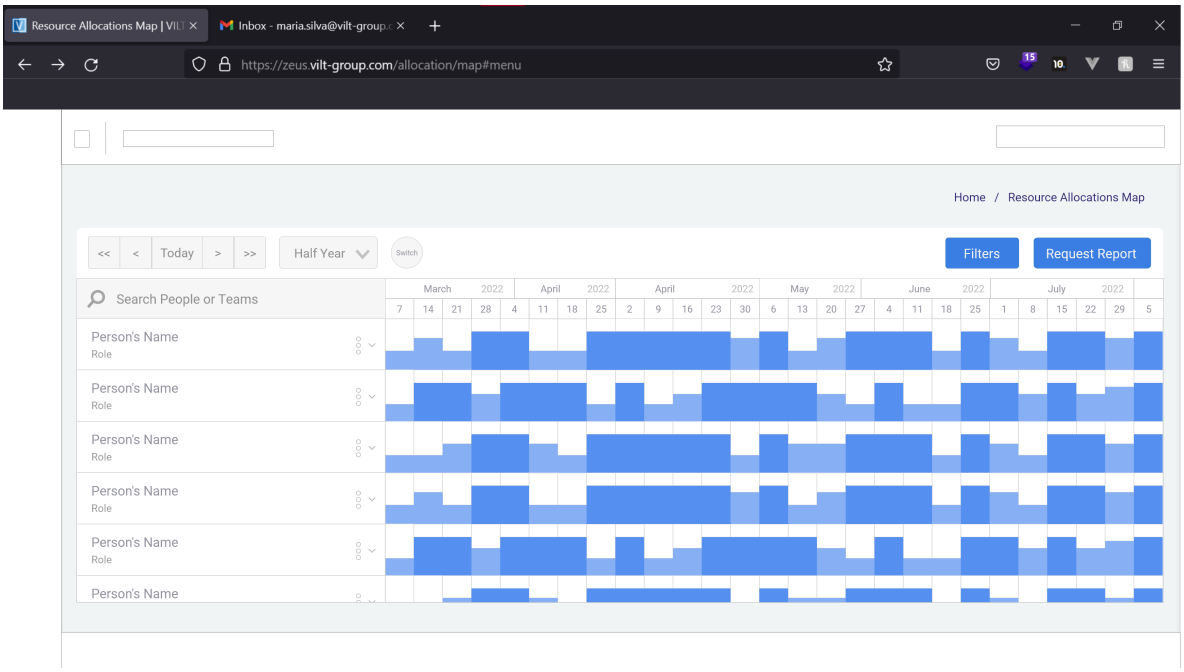

Figure 31: Half Year View

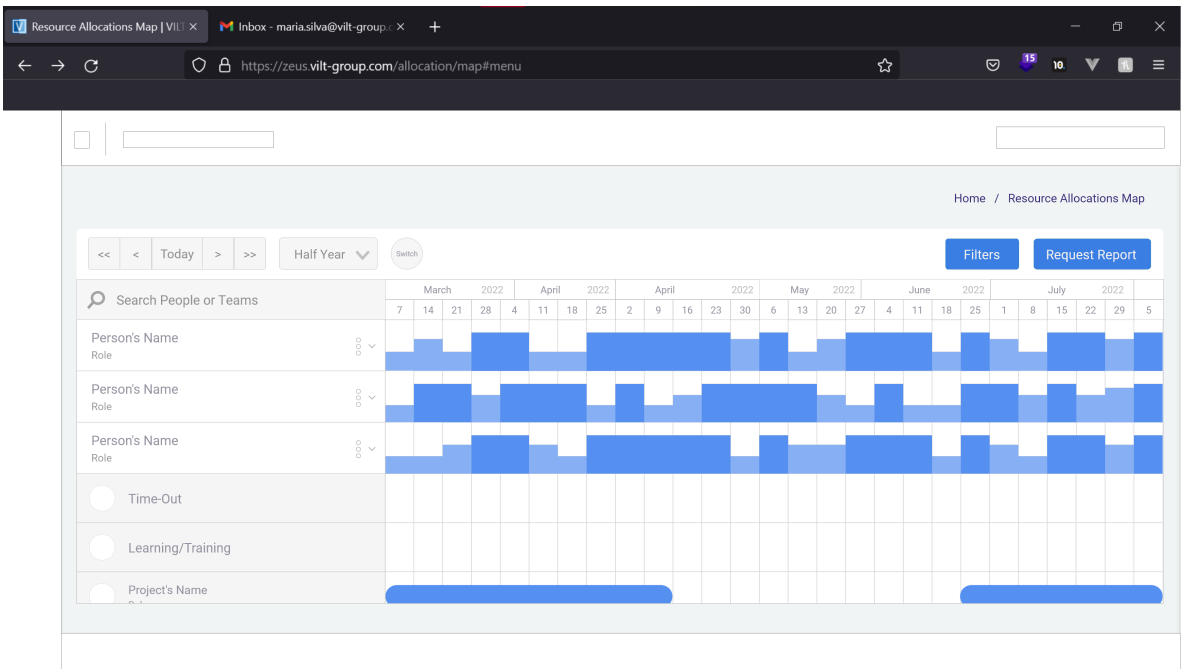

Figure 32: Half Year View - Open Row

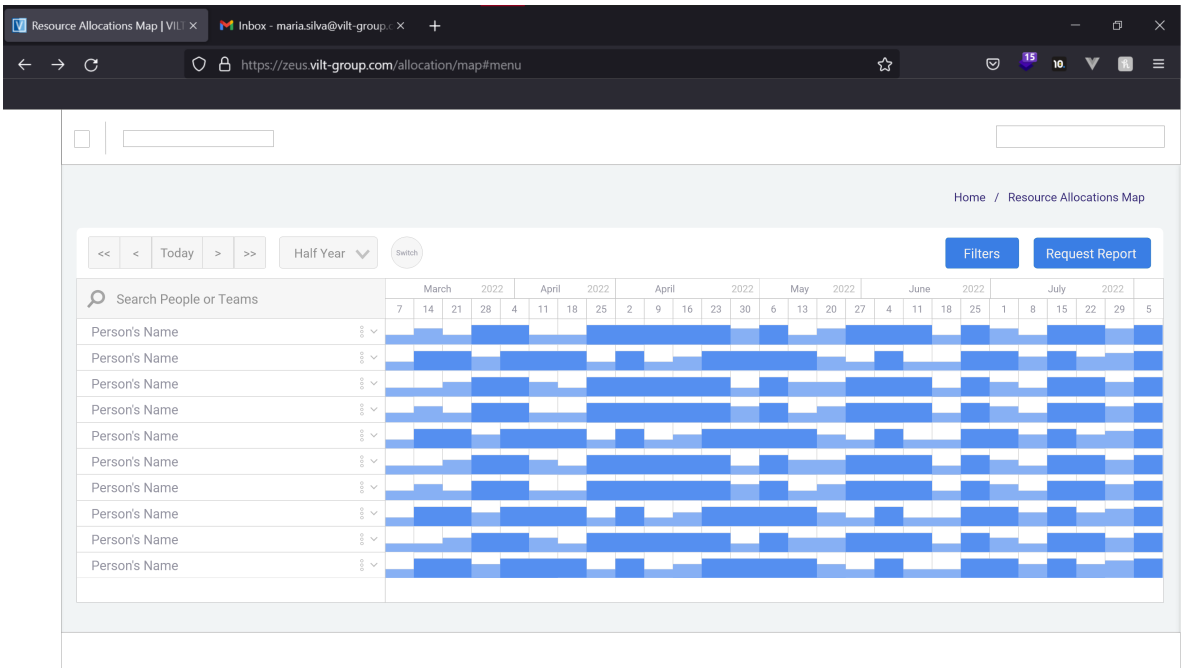

Figure 33: Half Year View - Compact

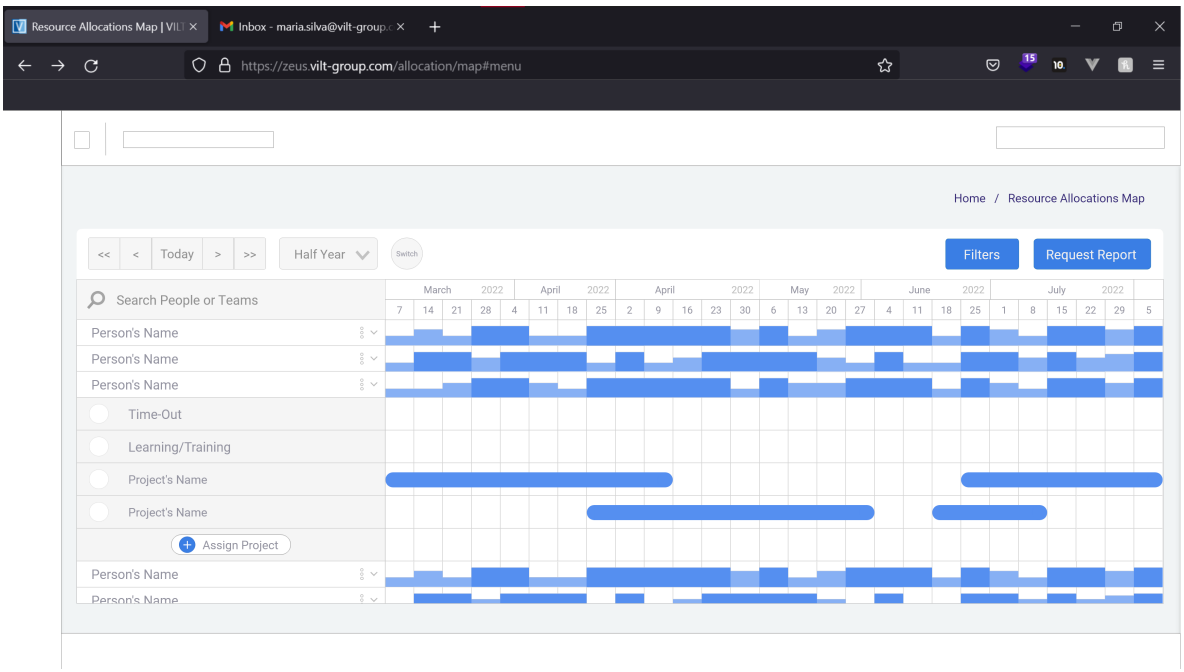

Figure 34: Half Year View - Compact - Open Row

We can see a big improvement when comparing the first prototypes and the final ones. The final prototype is interactive, created using *Adobe XD*, and was showed to the managers for final approval. It cannot show real data, since the objective was the design of the solution, but it can easily show the flow and interactions of the

final prototype. The tasks that are possible in the prototype are: scrolling through the rows; changing the view to month, quarter, or year; switch view to normal mode or compact mode; hovering over buttons to see a color change; clicking a row to open it and see the projects; and clicking on an allocation to see a pop up.

The final approval was obtained through a meeting with the supervisor managers. In the meeting, the prototypes were showed in action, and both managers gave their full approval for this to be the final prototype iteration. The following questions were asked after the approval, in order to get the final feedback:

- Do you think the prototypes are a good solution to be implemented? Why?
- Do you think the created solution is better than the old one? Why?
- Do you think this project was successful and met the specified goals? Why?
- Do you have improvements you would like to see in a future version of the project? Why?
- What is your overall feedback?

They believe the solution is ideal, since the gathering of the requirements was a complete process, taking into account different opinions and experiences. This resulted in a prototype that was designed according to the specific needs of the managers who will later use it, and that is visually appealing while being able to meet all the needs of the users. Comparing the new solution to the old one, they think the new allocation map allows a more intuitive way to view and manage information. As for additional improvements, they believe that in a future stage, after the implementation is live, and after a few months of usage, there will probably be things to improve, but nothing to suggest in this stage.

The creation of user support was considered. However, it was decided that user support would be best created in the future, after the solution is implemented. The developers can end up making changes, so the user support would not be as accurate now as it will be after the implementation of the solution.

4.6 PLUGINS

After the prototypes were finished and approved, an additional last step was done to help with the future implementation of those prototypes in a real usable solution. This step was the research of what plugins are in the market that can help build the project faster.

When comparing the different plugins, the following criteria were taken into consideration:

- **Free:** Is the plugin free or paid?
- **Collapsible Rows:** Can each row be collapsed?
- **Date Format:** Is the date format similar to the final prototypes?
- **Time Ranges:** Is there a way to include different time ranges?
- **Interactions:** Are there more interactions besides clicking on the bar or row?

After deciding the criteria to use, the following plugins were compared with the purpose of reaching the best decision to use in the future of the project.

- **jQuery.Gantt**

  This first option is the plugin already being used in the current solution (that we want to improve).

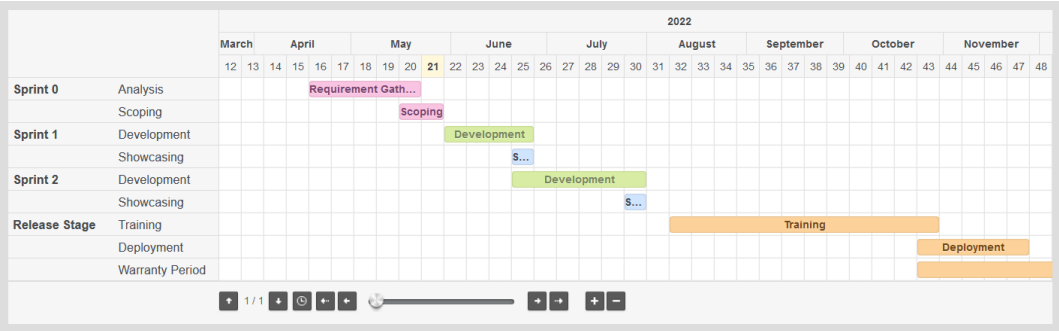

Figure 35: jQuery.Gantt Preview

- **jQuery.ganttView**

  This plugin seems similar to the previous one, with only a few design alterations. In terms of interactive action, they are really similar.

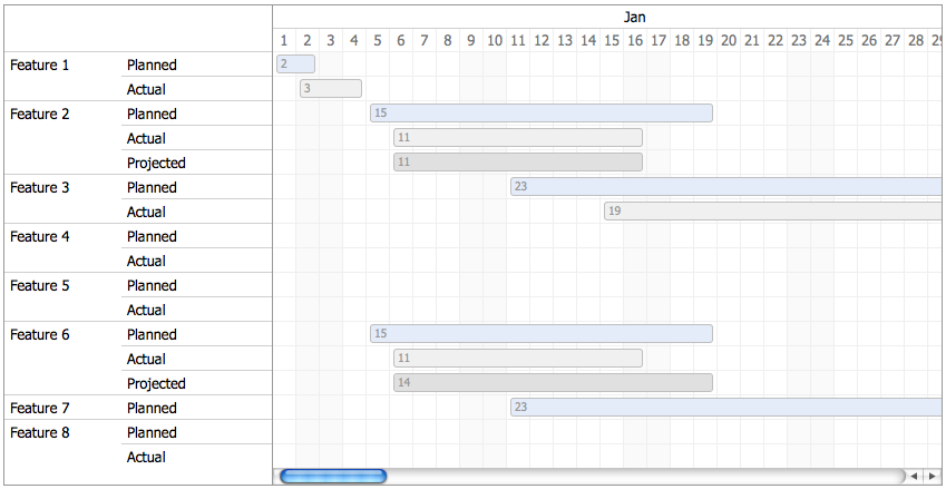

Figure 36: jQuery.GanttView Preview

- **AnyChart - JS Resource Gantt Chart**

  This plugin already looks more clear in terms of time visualization, and allows for rows to close, allowing the close view we had in the prototypes.

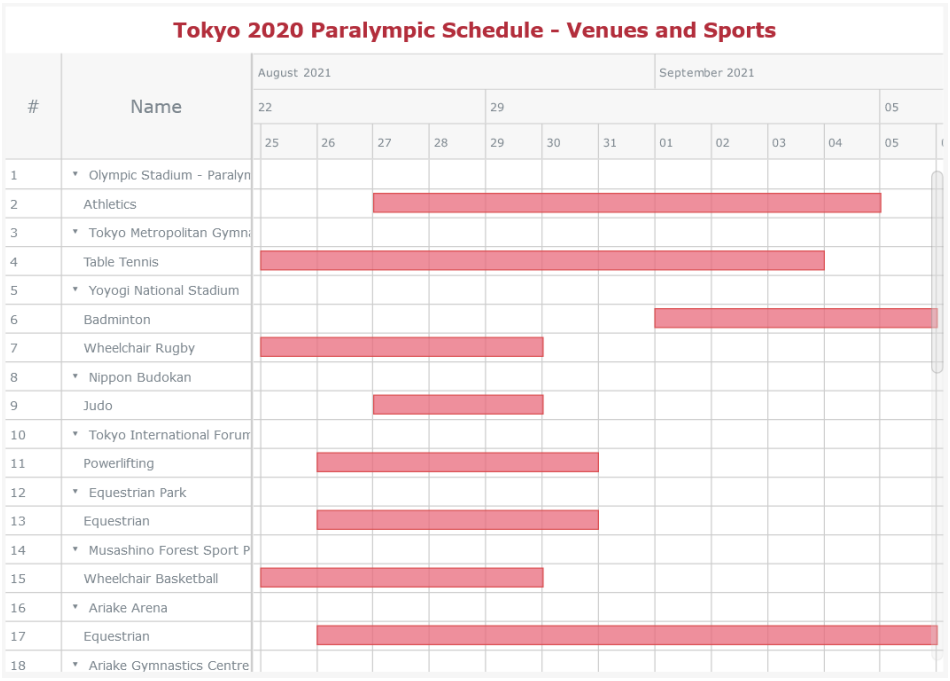

Figure 37: AnyChart Gantt Chart Preview

- **Webix - Gantt Chart**

  This plugin allows a big range of pre-created components and interactions, as well as the possibility to configure settings and personalize designs and tasks.

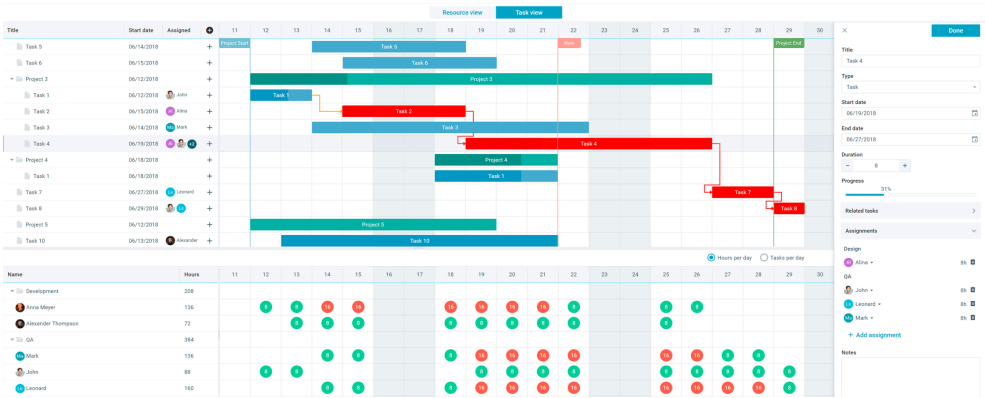

Figure 38: Webix Gantt Chart Preview

After considering the analysis above (see Table 2) , the decision was made to adopt *Webix Gantt*.

	Free	Collapsible Rows	Date Format	Time Ranges	Interactions
jQuery.Gantt	✓	✗	✓	✗	✗
jQuery.ganttView	✓	✗	✗	✗	✗
AnyChart Gantt	✗	✓	✓	✓	✗
Webix Gantt	✗	✓	✓	✓	✓

Table 2: Evaluation of the previous plugins

5

CONCLUSION

This research aimed to study the UX/UI Design process and different resource allocation solutions, and use those concepts to design a new resource allocation solution for VILT. This resource allocation solution (the allocation map) is a feature that allows managers to manage multiple human resources and their projects. A new solution was requested because the old allocation map faced limitations that made the managers at VILT choose alternatives.

The objective proposed in this dissertation was conducting a study that should take into consideration the managers' insights and ideas, inspiration from existing software, and inspiration from the managers' alternative solutions, in order to design an improved solution. It was desired that the new solution had an easy view of who is available and when, as well as the option to easily manage that information.

In order to achieve this, three smaller objectives were planned. The first planned step was doing user research and collecting the managers' ideas about what they liked and disliked in the old allocation map, as well as ideas for the new solution. The second planned step was creating prototypes and studying the managers' feedback until a final solution was designed and approved for future implementation. The third and last planned step was gathering feedback about the results and discussing it for future learning, as well as discussing future steps for when the implementation stage is realized.

5.1 ACHIEVED RESULTS

The objectives of this dissertation have been successfully achieved. UX and HCI concepts were researched initially, allowing a better understanding of the user through concepts about memory and cognition and its relation to the system. Different concepts and methods to design a solution were also researched, as well as how to test and improve the solution until the final version is reached.

After this, resource allocation concepts were studied, knowledge about Zeus was achieved, and some existing software choices were analysed for future inspiration.

Lastly, the learned concepts about UX, resource allocation, Zeus, as well as new knowledge acquired in this phase through user research, were used in the solution creation process. In total, three iterations of prototypes were done, having achieved full approval of the designed solution from the supervisor managers in the third iteration.

5.2 LIMITATIONS AND FUTURE WORK

The biggest limitation in this project was the reduced number of willing participants in the surveys realized. This could be avoided if the surveys were applied to a larger user base. However, since Zeus is an internal tool, it was desired that the surveys were realized only with VILT's managers. For this reason, there was difficulty in gathering enough people who had availability to spare working time. An interesting step to take would be realizing a bigger study in each iteration of prototypes, conducting a deeper and more time consuming study where the user could test the prototype in person, while saying feedback out loud.

The future plans for this project are the implementation of the designed solution, by a team of VILT's developers, using the study realized in this dissertation as a blueprint. Besides the implementation of the solution, it is also suggested the creation of a user support system to help the users with possible questions or problems. Another step for the future moment of implementation will be the choice of the available filters in the "Filters" button, which were not possible to define at this stage.

BIBLIOGRAPHY

[1] usability.de. What is usability? what is user experience?, n/d. URL https://www.usability.de/en/usability-user-experience.html.

[2] What is User Experience (UX) Design? The Interaction Design Foundation, n/d. URL https://www.interaction-design.org/literature/topics/ux-design.

[3] John M. Carroll. Human computer interaction - brief intro. In *The Encyclopedia of Human-Computer Interaction*, chapter 2. The Interaction Design Foundation, 2nd edition, n/d. URL https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-computer-interaction-brief-intro.

[4] Christopher D. Wickens, Sallie E. Gordon, Yili Liu, and John Lee. *An introduction to human factors engineering*. Pearson Education Limited, second edition, 2014. ISBN 1292022310.

[5] Gavin Ambrose and Paul Harris. *Basics Design: Design Thinking*. Fairchild Books, 2009. ISBN 2940411174.

[6] ISO. ISO 9241-11: Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts. International Organization for Standardization, 2018.

[7] Shawn Lawton Henry, Shadi Abou-Zahra, and Kevin White, editors. *Accessibility, usability, and inclusion*. Web Accessibility Initiative (WAI), 2016. URL https://www.w3.org/WAI/fundamentals/accessibility-usability-inclusion/.

[8] Jakob Nielsen. Usability 101: Introduction to Usability. Nielsen Norman Group, January 3 2012. URL https://www.nngroup.com/articles/usability-101-introduction-to-usability/.

[9] Jakob Nielsen. Enhancing the explanatory power of usability heuristics. In *Conference Companion on Human Factors in Computing Systems*, CHI '94, page 210, New York, NY, USA, 1994. Association for Computing Machinery. ISBN 0897916514. doi: 10.1145/259963.260333. URL https://doi.org/10.1145/259963.260333.

[10] Ben Aston. The 10 best resource management software & tools of 2022. The Digital Project Manager, Jan 2022. URL https://thedigitalprojectmanager.com/resource-management-software/.

[11] Anna Molly. Basic UI color guide. Prototypr, Dec 2016. URL https://blog.prototypr.io/basic-ui-color-guide-7612075cc71a.

[12] Cameron Chapman. Color theory for designers, part 1: The meaning of color. *Smashing Magazine*, May 20 2021. URL https://www.smashingmagazine.com/2010/01/color-theory-for-designers-part-1-the-meaning-of-color/.

[13] Aliza Ackerman. A guide to color meaning. Adobe, n/d. URL https://www.adobe.com/creativecloud/design/discover/color-meaning.html.

A

FORM QUESTIONS

The following questions were asked:

1.1. Do you actively use the Allocation Map in Zeus?

- Yes
- No, I use other methods

Only users who answered "Yes" to this question were shown the second section:

2.1. Why did you keep using Zeus' allocation map and not other methods?
2.2. What do you consider useful about the current allocation map?
2.3. Please explain your usual routine when using the current allocation map (step by step).
2.4. What do you wish was different in the improved version of the allocation map? What extra features would be useful to you?

The rest of the users were shown the third section:

3.1. What is the other method you use as replacement for the Zeus' allocation map?
3.2. Please explain your usual routine when using that substitute method (step by step).
3.3. What does the substitute method have that the Zeus' allocation map doesn't?
3.4. What would be useful to you in the improved version that currently does not exist in Zeus' allocation map or the substitute method?

Every user was shown the fourth section:

4.1. Do you have time to participate in an interview about these questions in the following week? If you choose 'Yes', I will contact you to schedule the date/time.

NB: place here information about funding, FCT project, etc in which the work is framed. Leave empty otherwise.