Sea & Surf

Hugo Miguel Carvalho Gaspar

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. João Manuel Brisson Lopes

Examination Committee

Chairperson: Prof. Daniel Jorge Viegas Gonçalves
Supervisor: Prof. João Manuel Brisson Lopes
Member of the Committee: Prof. Manuel João Caneira Monteiro da Fonseca

January 2021

Acknowledgments

I would like to first thank my supervisor Professor João Brisson Lopes for his contribution on this project. His constant availability to help, patience and constant incentive were without any doubt essential for the execution of this project with success.

I would also like to thank every professor and associate at Maretec for their great work on developing such fantastic models that help several people.

To all the users I have contacted for initial interviews and user tests, thank you for your time and collaboration. Your help was essential for the development of this project.

Last but not least, to my parents, brother, girlfriend and friends that helped me grow as a person and were always pushing me to end this chapter of my life with success. Thank you.

To each and every one of you – Thank you.

Abstract

The evolution of surf over the years has encouraged meteorologists to find new methods to improve predictions for weather and sea conditions. Nowadays surfers have all the information they need to know when and where are the best conditions to practice. Several websites and apps have emerged to display the data surfers need with some of them already being adapted to modern web technologies. This document reviews existing solutions and attempts to reach an optimized solution based on all the requirements needed. The solution takes advantage of the weather and maritime data that is already being modeled and retrieved by MARETEC. As an ultimate goal, it provides information to surfers in an efficient way and implements a responsive design that adapts the visualization of the information to the most common devices' size in the present days. In order for this project to stand out from the different solutions available on the market, a ranking for the surf conditions on each beach was developed. This ranking will allow surfers to interpret the data available in a easier way, since they will be able immediately identify if the beach is providing good conditions for surfing.

Keywords

Web Development; Responsive Design; Weather Forecast; Surf.

Resumo

A evolução do surf ao longo dos anos incentivou os meteorologistas a encontrar novos métodos para melhorar as previsões do tempo e das condições do mar. Atualmente, os surfistas conhecem todas as informações necessárias para saber quando e onde vão estar as melhores condições para surfar. Vários sites e aplicações surgiram para exibir os dados que os surfistas precisam, com alguns deles já adaptados às tecnologias modernas da web. Este documento analisa as soluções existentes e tenta alcançar e propor uma solução otimizada. A solução tira proveito dos dados meteorológicos e marítimos que já estão a ser modelados e guardados pelo MARETEC. Como objetivo final, esta aplicação fornece informações aos surfistas de maneira eficiente e implementa um design responsivo que adapta a visualização das informações ao tamanho dos dispositivos mais comuns nos dias atuais. Para que este projeto se destacasse das diferentes soluções disponíveis no mercado, foi desenvolvido um ranking para as condições do surf em cada praia. Esta classificação permitirá aos surfistas interpretar os dados disponíveis de uma forma mais fácil, pois poderão identificar de imediato se a praia oferece boas condições para a prática do surf.

Palavras Chave

Desenvolvimento para a Web; Design Responsivo; Previsões Meteorológicas; Surf.

Contents

List of Figures

List of Tables

List of Algorithms

Listings

Acronyms

HTTP	Hypertext Transfer Protocol
API	Application Programming Interface
JSON	JavaScript Object Notation
THREDDS	Thematic Realtime Environmental Distributed Data Services
NetCDF	Network Common Data Form
OPENDAP	Open-source Project for a Network Data Access Protocol
SPA	Single Page Application
GFS	Global Forecast System
MM5	Fifth-Generation Penn State/NCAR Mesoscale
WRF	Weather Research and Forecasting

1

Introduction

Contents

1.1 Context

Surf is a water sport where the practitioner, surfer, stands up on a surfboard and rides a wave towards the shore. This is a sport that is only possible whenever appropriate waves are formed in oceans, lakes or rivers. In order to know if there are favorable conditions to practice it, surfers need accurate information of sea conditions in real-time.

Information of sea conditions comes nowadays from forecasts, which are predictions of the future taking into account past and present data. This historical data is analyzed by specialists to identify trends and produce estimates for the future. Most of the time these estimates are produced in a way that can only be understood by specialists. Thus, with the evolution of the internet and with the emergence of mobile devices, several different solutions emerged to create visualizations that allowed users with no experience at interpreting forecasts to understand those estimates. Surfers can then take advantage of these visualizations to extract the information they need regarding the sea and weather conditions for the locations they intend to go surfing.

Although no forecast is 100% accurate, surfers can use the estimates for the different variables (wind direction, wind speed, wave direction, etc.) to have a better sense of how the sea is behaving. With this, they can better plan if it is worth to go to another beach or not worth to go surfing at all.

Of course, nowadays it is possible to have locations with artificial waves, but that case will not be studied since it falls outside the scope of this project.

1.2 Motivation

The previously mentioned forecasts are being produced by people and organizations from all over the globe to every location of the world.

The Maritime Environment and Technology Center, Maretec[1], located in Portugal is one organization that does research on weather and maritime forecast models. For weather forecasting, Maretec uses three different models (Global Forecast System (GFS) model, Fifth-Generation Penn State/NCAR Mesoscale (MM5) model, and Weather Research and Forecasting (WRF) model) to create predictions of the different weather variables (temperature, humidity, rain, etc.) for Portugal. The outputs of these models can be seen on a visualization in the website **meteo.tecnico.ulisboa.pt** that allows users to access it and get the information they need. Regarding maritime forecasts, Maretec has been developing MOHID, a three-dimensional water modeling system that allows modeling of almost all phases of the water cycle from watersheds to rivers, estuaries, and oceans. Thus, it is possible to conclude that Maretec is producing nowadays all the data needed to create weather and sea forecasts.

[1] http://www.maretec.org

Although the weather forecasts are already being displayed in the form of visualization on a website, the maritime forecasts are not. Therefore the main motivation for this work is to use the data that Maretec is collecting regarding both weather and maritime conditions and make it available in one visualization. More specifically, create a visualization adapated for surfers so they can know when and where are the best conditions to go surfing.

1.3 Main Goals

The first objective of this work was to analyze from the data that Maretec is already producing, which metrics are relevant for the practice of surf and which are not. Based on this analysis we can advise ways to make this information to be displayed in the most meaningful way so that surfers can make the most of it.

After the previous analysis, we moved to the main goal of this work which was to develop a website that enables users to access information in their own devices (smartphones, tablets or computers) in an effective way and having the same ease of interaction. It should be noted that conditions like devices' size and orientation, and sun exposure will influence the decision of the solution.

It is worth to note that the main feature of this application is a ranking of surf conditions that will be displayed for each beach location.

The end product is a solution for the visualization of information, from sea conditions forecasts, oriented to the practice of surf.

1.4 Methodology

This work followed a clearly defined and logical path which allowed us to sustain every decision made with evidences. The whole process can be divided in 3 big phases: Investigation, Development and Testing.

Regarding the investigation phase several different areas needed to be taken into consideration. Firstly, it was clear that we needed to gain some technical knowledge about surf and all of its components. Then, it was important to identify the users of our system and talk to them in order to identify their requirements. Thirdly, we needed to develop the theoretical foundation of our project, which consisted on focusing on the main concepts/principles a website should comply. Afterwards, existing solutions needed to be examined in order to identify their positive and negative aspects and consequently develop an optimized version of those. The last step of this phase was to study which technologies were the most suitable for this project in regards to performance and adaptability. Each of these steps were important in the development of the project because they helped us in making decisions regarding which path to take on every step of the process.

Regarding the Development phase, we needed to take some time to learn on how to use the technologies chosen, before it was time to actually develop the application.

Finally, the last step of this work was the testing phase which consisted not only on performing a functional evaluation of the system but also a system evaluation with the help of users.

1.5 Organization of the Document

This document has the following structure:

- **Chapter 1:** we provide the context, the motivation, the main goals and the methodology of this work.
- **Chapter 2:** we explore the fundamental components of surf, the optimal conditions for surfing, and finalize with a characterization of the surfers.
- **Chapter 3:** we provide a foundation for the decisions we have taken in the course of this project by reviewing principles of web design, analyzing existent solutions, by resuming the data and its representation, and finally by comparing existent technologies.
- **Chapter 4:** we present a detailed description of the solution developed.
- **Chapter 5:** we describe the methods we used to evaluate the solution and the corresponding results.
- **Chapter 6:** we conclude with some notes on the critical aspects of this work and with what can be done for future work.

2

Surf

Contents

Before analyzing the several solutions available for forecasting we must state some important information that underlines surf which is the main focus of this project.

First, we describe some of the structural aspects of surf namely the surfboard, the waves, and the environments. Next, we present the main conditions that surfers pay attention to when reading forecasts and deciding if there are good conditions. Lastly, we characterize surfers which are going to be the end-users of our solution.

2.1 Basics

In order to practice surf, there is the need to have a wetsuit but most importantly a surfboard, which is what lets surfers ride the waves.

The decision on which surfboard to use depends on the surfer, namely his/her weight, size, level of experience and preferred type of waves. It is possible to learn more about the characteristics and parts of a surfboard by looking at Appendix D.

Generally, for surfing, a good wave is the one that gradually evolves and breaks along the wall to one of its sides.

But before talking about what a good wave is in more detail, there is the need to understand the parts of a wave[1] (See Figure 2.1).

Face/Shoulder. Is the part of the wave that has not broken yet.

Peak. Is the highest point of a wave and where it begins to break. More experienced surfers read the peak to predict how the wave is going to behave.

Pocket/Curl. Is the part of the shoulder where the slope is steeper. This is where surfers perform most of the maneuvers since it is the point where the wave has the most riding power.

Lip. Is the upper part of a wave.

Tube/Barrel. Is the cylinder-like space of the curl that is covered by the lip. Most surfers use this space to ride inside the wave and perform a maneuver called tube riding.

Impact Zone. Is the place where the wave breaks and smashes onto the water. This is an area to avoid since it is where the wave is more powerful and can wipeout surfers.

Whitewater. Is where the wave already broke and foam is visible on top of the flat water. This is the place where beginners start learning how to surf.

There are 3 essential surf environments[2]: beach breaks, reef breaks, and point breaks.

Beach Break. Is the most common environment and it is where waves break on a sandy bottom. They are more unpredictable since the sandbars are constantly moving around with the tides. Generally,

[1] https://www.booksurfcamps.com/news/how-to-read-waves
[2] https://www.surfertoday.com/surfing/the-different-types-of-surf-breaks

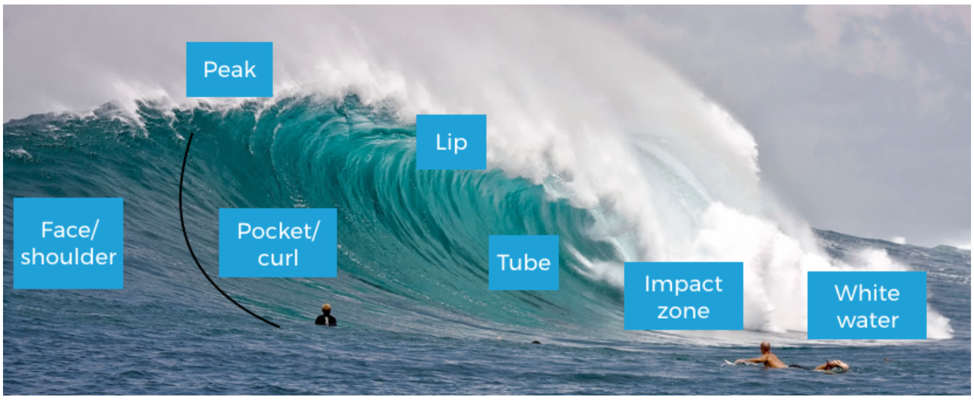


Figure 2.1: Different parts of a wave, source: boostyoursurf.com

this kind of breaks have better waves and are more consistent because they require less swell.

Reef Break. Is where the waves break over a reef or rocky bottom. The waves are more predictable in this environment since reefs and rocks do not move. However it is the most demanding and dangerous break since reefs and rocks can be extremely shallow in some areas and can hurt surfers.

Point Break. Is where the waves break alongside a headland into curved shorelines allowing for long rides. It is rare to find good point breaks since they depend on the swell angle and wind speed and direction to form good waves.

There are 2 types of wind conditions in surfing: onshore and offshore.

Onshore. The direction is from the ocean towards the shore. This is the worst one for surfers since it causes an uneven water surface which makes it harder to surf on. Nevertheless, a little bit of onshore wind is good because it helps surfers to keep their surfboards under their feet while performing maneuvers.

Offshore. The wind direction is from the shore towards the sea. This is the best kind of wind for surfers since it makes a smooth water surface which is easier to surf on. However, a strong offshore wind is not recommended since it can pushback surfers when performing aerial maneuvers.

Lastly, it is important to refer to a commonly used term in surf: the swell[3]. It is the energy that is transferred by the wind into the water. It is largely talked about by surfers since waves are formed by the strength of the wind over the ocean, thus the better the swell the better the waves.

2.2 Optimal Conditions

It is important to note that optimal conditions for surfing are subjective since they depend on the surfers' experience, style of surfing, location and kind of break. Surfers look forward to 4 main conditions when

[3]https://www.surfertoday.com/surfing/how-is-a-wave-formed-understanding-swell-and-surf-forecasts

looking at a forecast[4]: wave height, wind direction, wave direction, and the tide.

Wave Height. It is based on the swell height and period. The swell height is the distance between the flat surface of the water and the peak of the wave. The period is the time a wave takes to pass past a certain point. For example, a forecast showing a 2ft wave at 15 seconds shows that powerful waves will be formed since it has a 15 seconds period which means the wave is traveling from far away. If instead, the forecast showed a period of 5 seconds, it would result in weaker waves.

Wave Direction. This helps to predict how the waves are hitting a particular spot the surfer is looking for. For example, a location facing West with a swell direction from North would not be good since the waves would evolve to the left side of the beach, probably leading surfers to rocks which can be really dangerous.

Wind Direction. The best conditions for surfing normally require that the wind is barely existent. Most surf spots are calmer in the morning and evening and with stronger winds in the afternoon. Thus, it is important for surfers to check the type of wind (onshore or offshore) together with the spot they plan to surf to know if the wind is favorable or not.

Tide. Tides are dependent on the rotation of the Earth and the forces exerted by the Moon and the Sun, thus making it a predictable variable. The tide has a big impact on how waves break. Surfers need some experience to understand if it is better to have a low tide or a high tide for each spot. For example, in beach breaks, high tides are preferred since sand bars need to be deep for the waves not to break at once. While in reef breaks, low tide is preferred although it can be dangerous due to shallow rocks.

2.3 Users Characterization

It is essential to identify and describe who are the users and what are their demands to develop a system whose ultimate goal is to satisfy surfers' needs. Since we, the developers, are not going to be the end-users of the system this is an important step to be performed at the beginning of a study for an optimized solution. Users are the ones who know what information is needed and which is the best way to display it.

Thus we must know how certain users' characteristics might influence the way they use the system, how they learn to perform certain tasks, and how motivated are they to change habits. Overall, this analysis will influence the design solution to better address users' necessities.

In order to begin this study of users, we performed structured interviews with 4 different users (Appendix A). The interviews followed a specific set of short and concise questions that not only gave us the information we were looking for but also allowed some exploration of new ideas.

The following 11 questions help us to better describe and summarize who are the potential users

[4] https://learntosurfkona.com/featured/how-to-read-a-surf-report/

and what tasks they perform in certain environments [1].

1. Who is going to use this system?
    - Surfers.
    - Can be of any gender and any age group.
    - Can have previous knowledge of how to use weather forecast websites or none at all.
    - Have at least one device with internet access.
2. What tasks do they perform now?
    - Visit different forecast websites
    - Check the waves, wind, and tides of surf spots.
3. What tasks are desirable?
    - Access the conditions for today.
    - Access the conditions for the next 3 days.
    - History of the previous days.
4. How do they learn to perform the tasks?
    - When they are alone they can learn by comparing what they experienced at the beach with what the forecast predicted.
    - With experienced people (normally a surf instructor) by being told what are good and bad conditions.
5. Where are the tasks performed?
    - At home.
    - At work.
    - At the beach.
6. What is the relationship between the user and the information?
    - Users retrieve information from the table displayed in the web application.
7. What other instruments are available?
    - Beach flag (green, yellow or red).
    - Flags that limit the area allowed for surfing.

8. How do users communicate?
    - Word of mouth.
    - Text Messaging.
9. How often are tasks performed?
    - Daily if a frequent surfer.
    - Weekly if an occasional surfer.
10. Which are the time restrictions?
    - Users might take 1-2 minutes to perform a complete search.
11. What happens if something goes wrong?
    - User leaves the website and visits another one.
    - Attempt to communicate with other users.

From these interviews with the users we were able to define scenarios of activities that the users might perform in our system. As Fonseca, Campos and Gonçalves [1] state, two types of scenarios can help us in transitioning from the conceptual model to the first prototype. The first ones are called activity scenarios and describe users performing tasks to validate our solution in terms of how do they perform the tasks. It is possible to develop the following two activity scenarios:

- ”John wakes up in the morning and it is a beautiful day outside. Since it is Summer, and he has no classes, he decides that he wants to go surfing to his favorite beach, Guincho. He unlocks his phone and goes to one website that provides surf forecasts and checks the conditions for Guincho. Fortunately, according to the website John checked, the waves at Guincho are incredibly good. He gets his wetsuit and surfboard and drives to Guincho to catch some wonderful waves.”
- ”Michael is a surf instructor. He has three students that are really committed to learning how to surf, and this week have booked classes with Michael every day. Today is the third day they are having class and Michael decides to start teaching them how to read and understand a surf forecast. He opens his smartphone and goes to a surf forecast website. Since Monday was when there were worse conditions for surfing he starts to select the beach they were on Monday and made his students verify that actually, the forecast said that waves were a little small for that day. He then proceeded to show that on Tuesday the height of the waves drastically changed for better and they saw it at the beach.”

The second type of scenarios are called interaction scenarios and describe users performing their tasks using a specific design for the interface. These are normally based on the activity scenarios and describe the physical actions performed by the users over the elements of the interface and the corresponding response of the interface. This allows us to better imagine and design the interface. The following interaction scenarios can be developed:

- ”John wakes up in the morning and it is a beautiful day outside. Since it is Summer, and he has no classes, he decides that he wants to go surfing to his favorite beach, Guincho. He unlocks his phone, opens his favorite browser and goes to Sea&Surf website. He clicks on the top search bar and types the name of the beach he wants to go, Guincho. The website shows a loading button that lasts for 1 second. Then, John is faced with a map of Portugal centered and zoomed in at Guincho. The map has a color scheme that is now representing the wind intensity over that region and has some arrow glyphs showing the direction of the wind. A row is displayed at the bottom of the screen showing information regarding the waves and tides for Guincho. Fortunately, according to the Sea&Surf website, the waves at Guincho are incredibly good. He gets his wetsuit and surfboard and drives to Guincho to catch some wonderful waves.”

- ”Michael is a surf instructor. He has three students that are really committed to learning how to surf, and this week have booked classes with Michael every day. Today is the third day they are having class and Michael decides to start teaching them how to read and understand a surf forecast. He opens his smartphone and goes to the Sea&Surf website. Since Monday was when there were worse conditions for surfing he selects at the top search bar the beach they were at. Below the search bar, there is a timeline where he then selects the previous Monday. The system shows a loading screen and then prompts with the conditions for that Monday. Michael now shows the interface so his students can check that actually, the forecast said that waves were a little small for that day. He goes back to the timeline and now selects the previous Tuesday. The loading screen appears again, and a second later students can check that the forecast predicted that the height of the waves would drastically change for better and they saw it at the beach.”

Both these types of scenarios help us in verifying problems with our solution and serve as guidelines to define the tasks to be performed on the tests with users on the evaluation phase. Since these scenarios describe users performing their tasks we are able to realize if those tasks are too complex to be performed and need to be simplified.

As stated before, certain user characteristics influence the design solution. When we were talking with users we realized that there are 2 user characteristics that need to be taken into account:

Users with vision problems. Since this is a condition that some users might have, smaller objects and text should be carefully examined before being used on the solution.

Color-blind users. Since we might have surfers that can not distinguish some colors, color palletes should use a scheme of colors that address these difficulties.

From the interviews it was also possible to notice that there is an environment characteristic that will influence the design, which is the glare. Since surfers mentioned that sometimes they consult forecasts at the beach, which is an environment subject to bright sunlight, we must create high contrasts in our interface so that different elements can be seen.

Another aspect of the characterization of users is their motivation. Some users might be willing to learn and change to another website while others might not be interested in changing. Users play a huge role on the overall success of the system and therefore they should be examined and consulted as much as possible. We can say that a critical success factor of this work is the users' satisfaction and acceptance of the end product [2].

In regards to users' satisfaction, Muneera Bano and Didar Zowghi [2] define it as a "psychological state that comes when users perceive that they have control over the system development process". This leads us to other activity developed with the users: sketches.

In Figure 2.2 we see a sketch made with the help of a user. By involving the users in the creation process we did not only increased their motivation but also defined a good starting point for the development of our interface.

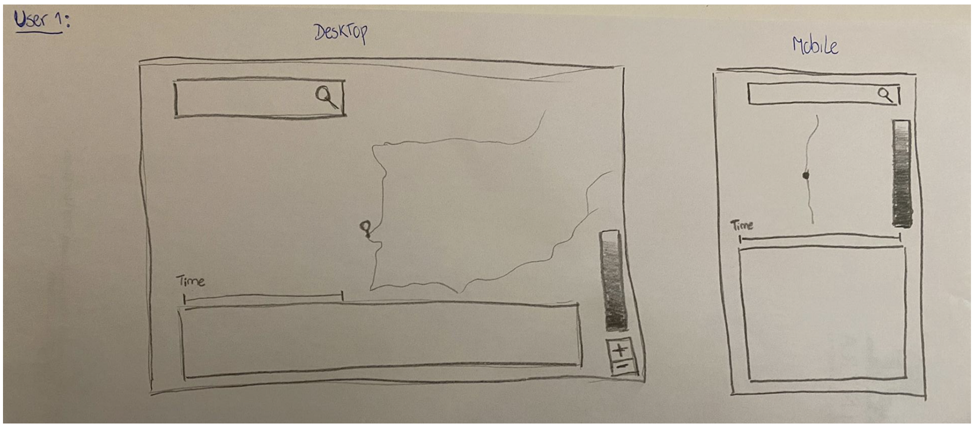


Figure 2.2: User Sketch

With this analysis, we addressed the main aspects that will help in the development of a system centered on the user. In a later section, we are going to detail what was done in order to address each of the users' needs and characteristics. Also, the users will be referred once again in the evaluation chapter, where we include them in the user tests performed on the system.

3

Concepts, Principles and Existing Solutions

Contents

The main goal of this chapter is to establish a solid foundation to backup the developed solution that is going to be presented in the next chapter. For that, it is important to focus on some of the key principles that every web application should respect. Then, we shall evaluate some currently available forecast websites based on the previously described principles. Afterward, we detail the meteorological and hydrographic data needed for this kind of forecast visualization. Lastly, we compare existing technologies, so we can further explain the reasoning behind the choices made.

This analysis together with the users' characterization in the last section will help explain how our solution is centered on the user, respects the principles of web design, and ultimately takes advantage of the strong features and fixes some weaknesses of the existing websites.

3.1 Principles of Web Design

To facilitate the interactions users might have with our website we must develop a good design and provide a good user experience. There are lots of principles and guidelines to follow when building for the web but we are only going to focus on five: responsive web design, effective homepage, minimalist design, interactivity, and navigation. This decision was based on what we considered the most important according to the several guidelines that Manuel Fonseca, Pedro Campos, and Daniel Gonçalves presented in "Introdução ao Design de Interfaces" [1] as good practices for the web.

3.1.1 Responsive Web Design

Responsive Web Design (RWD), also called Adaptive Web Design, is a set of approaches that programmers and designers should apply when building websites. It focuses on having a website that is capable of resizing itself depending on screen size, resolution and orientation of the device that is being used to access the website [3].

To build a responsive website two strategies can be used depending on the existing context. The first one is called graceful degradation and is used when the website already exists and needs to be redesigned. In this case, the website is normally designed for large devices so there is a need to reduce the number of visible elements and rethink the organization of the elements to have a nice and clean website in smaller devices. The second strategy is called progressive enhancement and is used when developers are creating a new website and start from scratch. In this case, the website is designed for smaller devices first because it is better to first reduce the information to the essentials and then gradually scale the design for larger devices, where it is much easier to organize the information since there is more viewing space available [4].

To have a responsive website there are 3 main aspects to follow [3] [5]:

1. **Fluid Layouts.** Websites are developed using a grid-based layout that uses HTML tables and divs. RWD should be applied to this grid in a dynamic way instead of using a fixed width. The developer should specify the size of each component of the website based on window percentages and not pixels. This allows for the resizing and reorganization of the website that will be viewed based on the percentage ratio of each element relative to the viewing device's dimensions.

2. **Flexible Images and Media.** Media objects like images and videos have a default size which will prevent the layout to be reduced below the media's width. To fix this issue there is a need to scale the media object according to its parent container. This is done by using the max-width CSS property on each object which enables the browser to reduce its size in order to fit in the container's size without adding scrollbars.

3. **Media Queries.** Media queries were introduced with CSS3 to allow browsers to show different versions of a website for different viewing contexts. By specifying the different resolutions of the screen, developers can organize information in a different way and make the website more user-friendly. It is not feasible to target all possible screen resolutions and create a different design for each. It is then needed to analyze the most currently used screen resolutions and group the remaining accordingly.

3.1.2 Effective Homepage

A homepage is the most important page of a website because it is the one with the most affluence of users. A well thought and designed homepage should satisfy the following requisites [1]:

1. **Elicit a good first impression from users.** If this first impression is positive, users might want to stay and navigate through the website and come back later. Otherwise, users will leave in search of alternatives.

2. **Facilitate navigation.** This is the place where users can find easily and quickly what they are looking for. Even though some users might not start at the homepage, either because they are a returning user and know exactly where to find the page they are looking for, or because when searching, the best result to their search query is another page on the website, they will eventually look for it if they need to search for something else on the website. Thus, an organized home page with a clearly defined main topic is fundamental to help users find what they are searching for.

3. **Focus on the main topic.** When a user reaches a homepage, he/she needs to know where to start. The main element of the page should be representative of the website and stand out from the rest of the elements.

3.1.3 Minimalist Design

Having a minimalist design is a way to facilitate the identification of the relevant parts of a webpage. Since the web is not a static environment there should be an even balance between aesthetics, functionality, and usability [1].

When developing a website, each element should be considered and if it is not required to support the core functionality of the website, it should not be included. Therefore when talking about a minimalist website, we are referring to a "clean" website that reduces its elements to the essential ones and provides no distractions to the user [6].

Some characteristics distinguish a minimalist design:

1. **Restricted elements to maximize negative space.** By removing non-essential elements from a webpage we leave empty space. This space can either be white or have any other color or texture depending on the interface. The main purpose is that we can direct the user's attention to what really matters on a webpage.

2. **Flat rather than skeuomorphic patterns and textures.** In the early stages of web development, skeuomorphism was used to help people in associating digital elements with physical ones. Nowadays, most users have evolved and can visualize digital elements and consequently understand them without the need for metaphors. Thus, using highlights, shadows, gradients or other texture features are considered unnecessary. A minimalist design is one with a simple style, two-dimensional elements, and bright colors.

3. **Thoughtful use of typography to convey meaning.** By applying changes to font weights or sizes it is possible to create interesting features without the need to add more design elements or graphics. When eliminating non-essential elements, and to avoid the risk of having a boring interface, we can explore different typography to compensate it. Also, those variations can help users understand different hierarchies and pay attention to more important elements.

4. **Use of a limited color palette.** Since in minimalist designs the quantity of information is reduced, color palettes are more noticeable and have more influence on a site's impact. To highlight and contrast different elements of a webpage, variations in color hue and saturation are normally applied instead of adding different colors.

5. **Use of a grid layout.** This is directly related to one of the previously mentioned principles of web design (RWD). By using grids it is possible to better organize content and to better communicate with users without adding more elements.

3.1.4 Interactivity

Interactivity is a key feature of websites since it is what makes users to be actively engaged while navigating on a website.

According to research made by Yang & Fuyuan [7], ”web interactivity could lead to increased user enjoyment, positive attitudes, and favorable behavioral intentions”. Having interactive features such as map flows can not only make the experience of the website more fun but also present the information in a more meaningful way as well as give more control to users. Also, users are more likely to come to revisit and recommend interactive websites, since they feel more intrinsically motivated to perform tasks.

However, offering too much interactivity may distract users from performing their tasks, which can then lead to negative effects like cognitive overload and frustration.

3.1.5 Navigation

The structure of a website and the navigation inside of it are two fundamental aspects of web development. Normally, websites have a hierarchical structure that helps the user when searching for specific information. This way, the information is organized and lets users immediately find and go to the section of the website that they are looking for.

Most websites use menus at the top of the page to facilitate navigation and show the main services provided by the website. These menus take users to those pages or to submenus of those pages. Users can then use these menus to find what they are searching for and go to specific pages of the website.

Also, it is important to know where we are when navigating on a website. Web designers must provide clues about a user’s location on a website, so they don’t get lost and can proceed effectively. Four different resources can be used to provide guidelines on users’ location on a website [1]:

1. **Page title.** Must clearly indicate the website and the page users are on. It should be short and vary from page to page.

2. **Use separators.** Change the color of the currently selected menu is a good way to provide the location. Also, using different colors to group menus can help better understand the organization of the website.

3. **Breadcrumbs.** Showing the path that the user took to get to a specific page not only shows the current location but also provides a simple way to return to previous pages.

4. **Home Button.** Having a home button located on the top of a page provides a simple way to lost users to return to a known place of a website.

3.2 Forecast Websites

Having the guidelines defined for evaluating the websites in regards to the principles of web design, we are now going to perform a detailed analysis of some websites. There is a considerable amount of forecast websites available on the internet, but for this analysis, we are going to focus on only three websites: Windguru, Windy, and MagicSeaWeed. This selection was made taking into account two factors: popularity and different design solutions. By looking at the answers from the interviews (see Appendix A) we can see that Windguru is the most used website by surfers. Windy and MagicSeaWeed are other websites used by interviewees that were chosen because of the different design solutions they present.

Although the focus of the evaluation will be on the principles earlier defined, we will also describe specific features of the different solutions if they are worth to note.

3.2.1 Windguru

Winguru[1] is a service specialized in weather forecasts mostly for windsurfers and kitesurfers. Nonetheless, as we found out from the interviews, many surfers use the information displayed on this website to know if there are good conditions to go surfing.

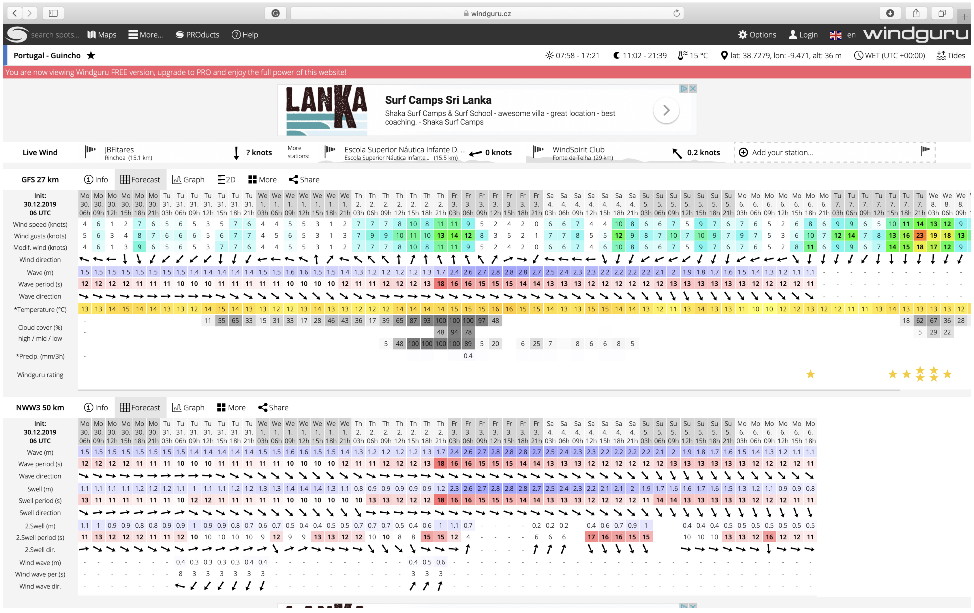
Based on the previous list of principles of web design we are going to analyze Windguru and note if those principles are being respected and, if not, where they are failing.

This website does not respect the first principle previously mentioned. Comparing the website on mobile devices with the website on desktops (see Figure 3.1), we see that the information is being displayed in the same way. No actual reformulation is done to the presentation of the information when viewed on smaller screens, making it really hard to interpret the data.

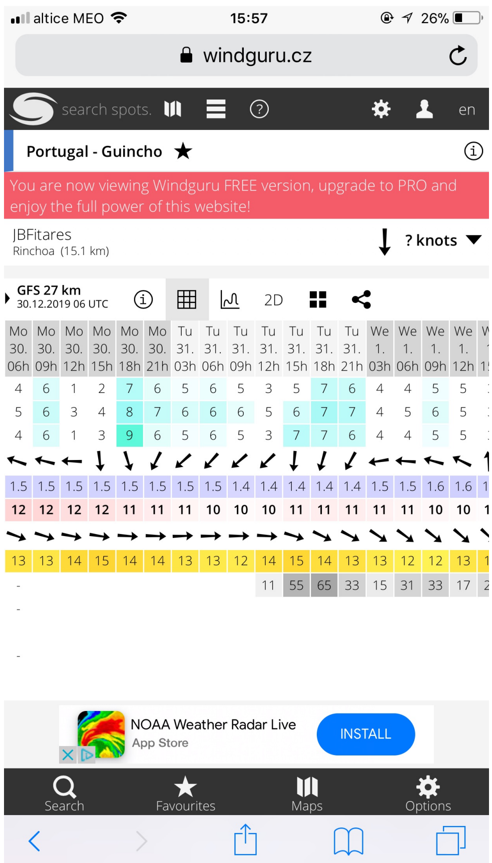
When entering the website we are immediately faced with loads of information and a lack of organization. The page loads information regarding a specific beach without the user choosing one (most of the time it is a beach located in a different country from the user). Overall it is not a pleasing experience since there is too much information showing up at the same time and it is difficult to understand how to access the desired information. It is evident that this website lacks a proper homepage since it does not create a good first impression for users and does not facilitate navigation. Thus, the second principle previously stated is also not respected.

Moving to the third principle, the following aspects make us realize that the design does not follow a minimalistic approach. When viewing the forecast for a specific spot we are faced with several different weather models, most of them unnecessary for the majority of users. It would probably be better to show the information provided by the most common model (i.e. the one the majority of users would use) and

[1] https://www.windguru.cz/



(a) Desktop



(b) Mobile

Figure 3.1: Windguru Homepage

make available a select option where the user could change the model. This would reduce significantly the amount of information that is being displayed.

Another thing to note is the excessive use of advertising (See the yellow marks around the ads on Figure 3.2). The website is already pretty loaded with information and then users are shown several ads occupying a considerable part of the screen. These ads interfere with the website's design and content, worsening the user experience and probably making users want to leave the page.

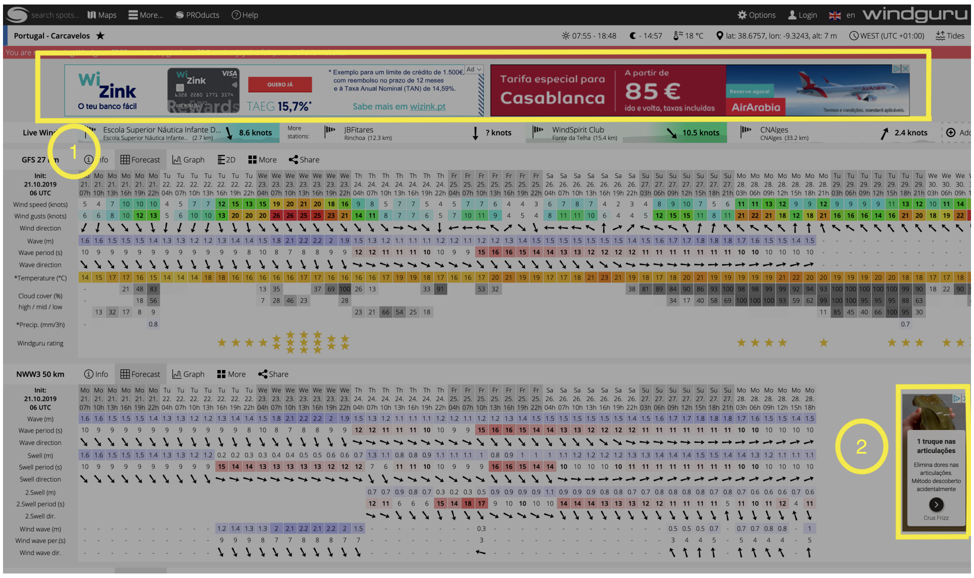


Figure 3.2: Advertisement shown in a Windguru page.

Overall the website is too crowded and therefore does not have a clean design that takes advantage of negative space.

In relation to the fourth principle, we can say that the website allows for some interactivity. Besides forecasts, it is possible to generate graphics that show the variation of the different weather variables through time. Users can also click on specific values of the wind to open small windows showing the behavior of the wind for the area surrounding the selected beach. By allowing this interactivity Winguru is definitely improving the experience of its users.

Regarding the fifth principle, although the website provides a clear navigation menu on the top of the page, some of those menus lead to unavailable pages with a ”coming soon” message. These pages are blank and force users to go to the old website if they want to check that content. This bad practice makes navigation difficult by displaying categories in the menu that aren't available on the website and therefore lead people out of it without an intuitive way to come back. Also, in the majority of the webpages, there is no information provided regarding the user's location on the website either through titles, separators or breadcrumbs. In sum, the navigation on the website is not well-conceived.

To conclude this website has two additional aspects that are worth to note. The first one is that users can access all sorts of information regarding weather forecasts and sea conditions. Windguru presents a variety of weather models that predict the state of the atmosphere for the next days. These models use different equations, resolutions, and combinations of initial data sources to produce forecasts. This is a good feature for experienced users that know how to interpret the different models and take advantage of them.

The second one is Windguru rating. Although it was developed for windsurf and kitesurf, this rating uses star count to convey information to the users of how good is that spot for the practice of sports, meaning that the more stars it has the better the conditions. This kind of information is really helpful to add more meaning to the information that the website shows. Also, it is a big assistance to people that are not good at interpreting forecasts.

3.2.2 Windy

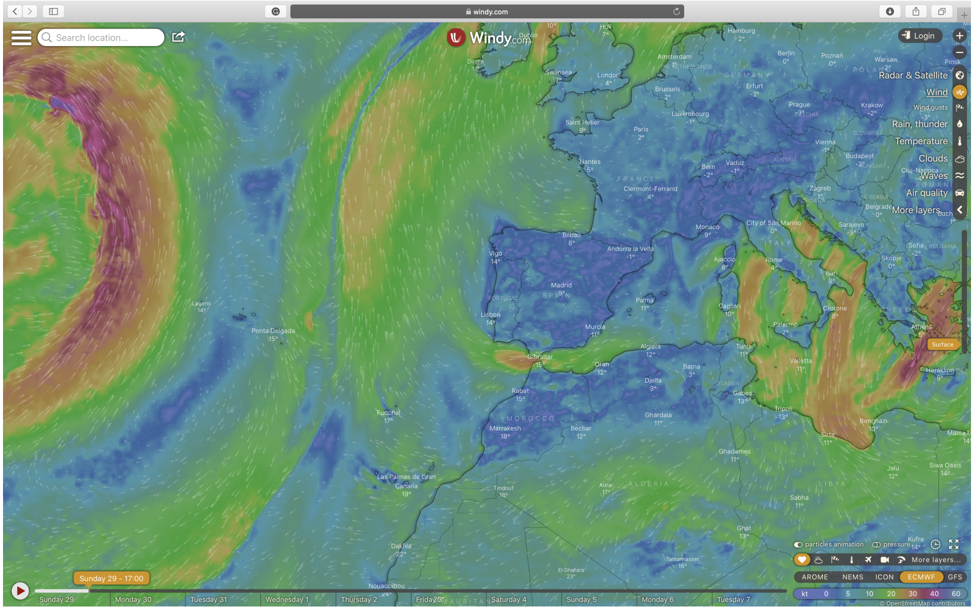
Windy[2] is a weather service that provides worldwide weather forecasts since 2014. As the website name suggests the main focus is the wind and its different components. Nevertheless, it shows all kinds of information we look for when looking at weather forecasts (rain, temperature, etc.).

A good starting point to argue is that this website follows the principle of responsive web design. By looking at Figure 3.3 we can compare the homepage in a desktop with the homepage in a mobile device and see that there is a different arrangement of the information. In mobile devices, the search bar has a bigger highlight by occupying the top part of the screen and we are shown a smaller area of the map. This might be explained by the fact that it is harder to select the exact location on the map on mobile devices so by zooming in and highlighting the search bar we are helping users to get what they need. Also, it is possible to note that the menus are more condensed and organized in smaller screens.

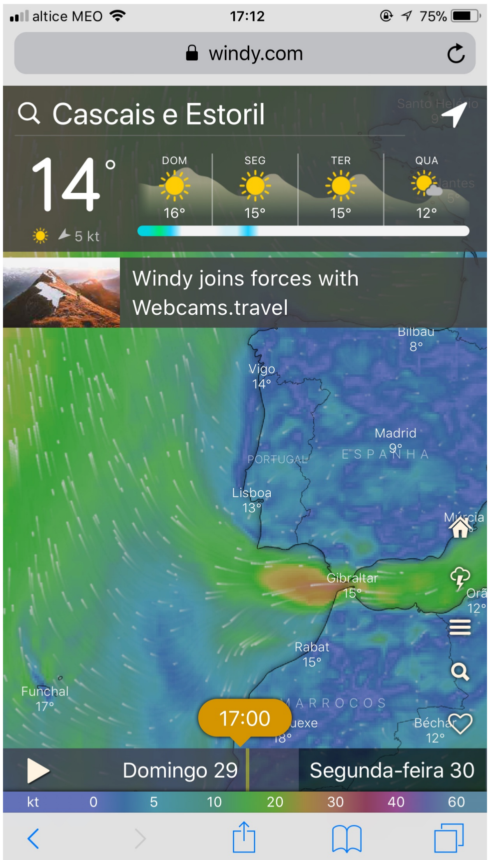
Windy is a single-page application (SPA), which means that it only uses one page that is re-rendered using Javascript whenever users interact with it. Instead of loading new pages from the server this website loads and removes elements from the page based on the user's requests. So, in this case, there is no homepage but instead a single page. Nevertheless, this page should follow the requisites we have defined for the homepage. Overall this website has a nice design eliciting a good impression from users, facilitates navigation by providing several menus and options where users can find what they are looking for. The most important thing to note on this website regarding this principle is that we know immediately which is the main topic by being presented with a map that occupies the majority of the screen size. Initially, this map is centered on our location and shows the main variable they focus on, which is the wind. Therefore, we can say that the second principle is being respected.

Regarding the minimalist design, we can say that this website has its elements reduced almost to the essential ones when compared to Windguru. When we look at the overall page we can see that all

[2] https://www.windy.com/



(a) Desktop



(b) Mobile

Figure 3.3: Windy Homepage

the present elements need to be there to give the information users typically need in a forecast website. By having only the necessary elements, the website has more negative space that is being used to increase the size of the map instead of just having a blank background. The only thing that we can argue against this website regarding this principle is the excessive number of colors used. In this case, as we are representing several variables and the ranges of their values are quite big there is the need to have more colors to build a suitable and effective scale. But instead of adding more colors we could make variations in color hue and saturation.

Windy provides a good amount of interactivity to users. By clicking on a specific spot on the map, the website automatically describes the name of the place or its coordinates (depending on if it is land or ocean) and gives all sorts of information for that specific location. Also, as seen in Figure 3.6(b), for variables that have a direction component such as wind or wave, users can activate the animation to show those directions on the map. These are examples of features that help users to better understand the information that is being displayed consequently giving them a better experience on the website.

With regards to the fifth principle, since once again this is a SPA, we do not have navigation between pages. Nevertheless, we have menus organized on the right side of the screen that allow us to change the variables that are being represented on the map. This website highlights with a yellow background the options that are selected at the moment, helping the user to know what he/she is viewing.

Lastly, this website has a feature that must be taken into consideration which is the symbols used. Most of the menus and variables of this website have a symbol associated with it which replaces the text when there is less space to represent elements. These symbols help users by communicating the information needed in the easiest way possible. On Figure 3.4 it is possible to see some symbols that are used on this website.

(a) Temperature **(b)** Sea Temperature **(c)** Humidity **(d)** Wind **(e)** Waves **(f)** Swell

Figure 3.4: Symbols used by Windy to describe variables.

3.2.3 MagicSeaWeed

MagicSeaWeed[3] was developed more than a decade ago and creates meteorological and ocean forecasts specifically for the practice of surf. Although only one interviewee used this website we chose to analyze it because it mainly focuses on surf which is the main focus of this work and because it has an overall different structure when compared to Windguru and Windy.

When looking at the homepage in desktop versus in mobile (See Figure 3.5) we notice that responsive design is being applied. While viewing in a desktop the menus are all shown from the start, in mobile, a dropdown is being used to show the menus whenever a user clicks on it. Also when moving from desktop to mobile it is possible to check that images have a flexible width depending on the parent's container. Overall there is a different arrangement of the elements when comparing the website in different screen sizes.

MagicSeaWeed's homepage is a good example of what a homepage should contain to be effective. When we enter it we can immediately understand that this website is about surf. Then, by taking a look at the menus at the top of the page users get their first impression of what to expect to encounter on this website regarding surf. Since it is well organized and clearly defines what are the several menus available it facilitates user's navigation on the website.

Regarding the third principle, this website has the same problem as Windguru. In almost every page of this website, we are faced with ads that occupy a considerable amount of the screen size. Also, in some pages, we can see that there are more elements present than the necessary ones. For example, if we want to check the forecast for a specific Portuguese beach, we have on the same page beside the forecast, other elements like the webcam, news or products to buy, that should be in the correspondent menus. For these reasons, this website does not have a minimalistic design.

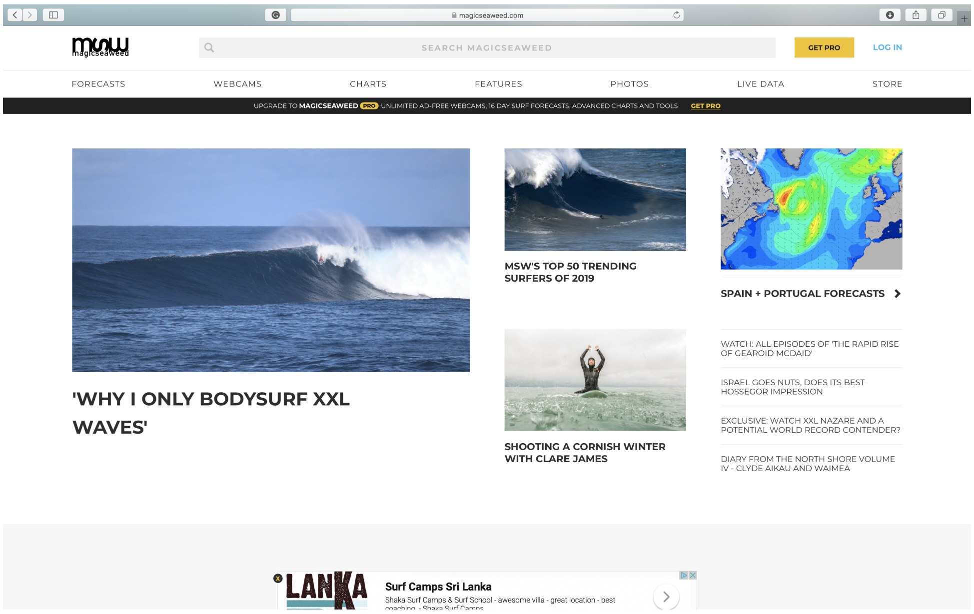
Moving to the fourth principle, this website provides limited interaction with it. Although users can select which weather model, units, time and location they intend to see, there is not much more they can interact with. After selecting the location users are presented with a page that looks like a paper report since it does not allow for changing the kind and organization of information displayed.

Lastly, the navigation on this website is good. As already mentioned the menus at the top help users to find what they are looking for easily. Also, by displaying the title of the page, changing the color of the selected menus, and having a home button, users can always know their location on the website.

3.3 Data

Taking into consideration the previous three websites and the interviews with surfers we can create a list with the meteorological and hydrographic information that is needed for such visualizations. Table 3.1

[3] https://magicseaweed.com



(a) Desktop

(b) Mobile

Figure 3.5: MagicSeaWeed Homepage

details the attributes and respective units that are essential for producing a surf forecast website.

Table 3.1: Meteorological and Hydrographic data and corresponding units.

	Magnitude	Units
Meteorological	Temperature	ºC
	Wind Speed	km/h
	Wind Direction	Cardinal Coordinates
Hydrographic	Water Temperature	ºC
	Wave Height	meters
	Wave Period	seconds
	Wave Direction	Cardinal Coordinates

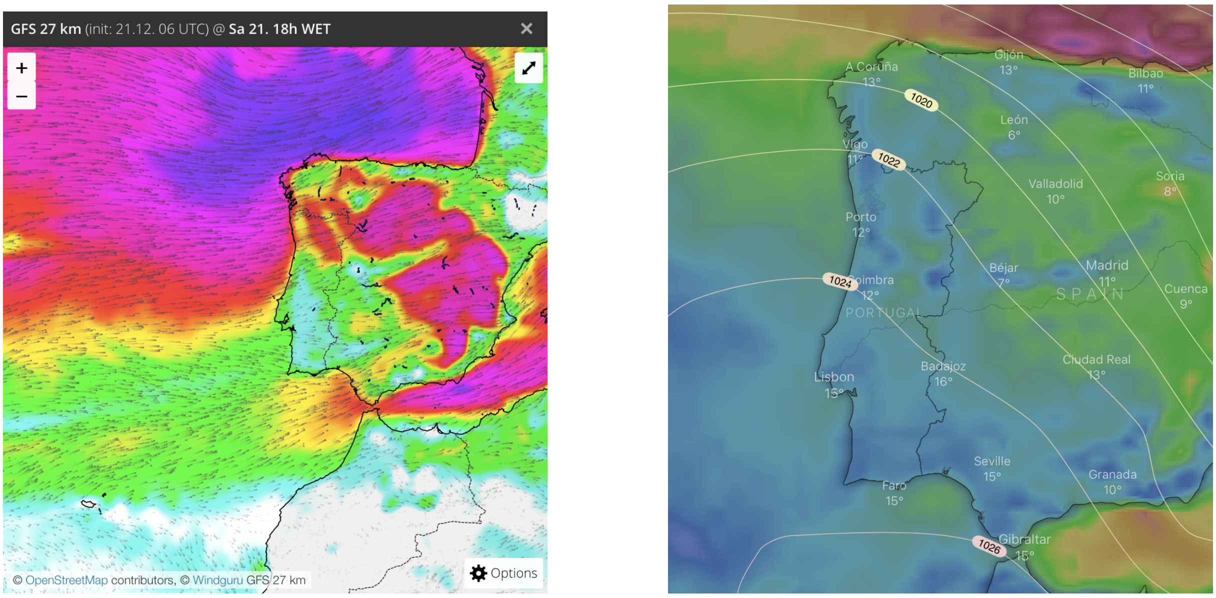
Now that we have carefully addressed the most important data to collect, we need to understand how it is represented in the visualization. The previously analyzed solutions take advantage of three ways to represent data:

Color. The use of color is one of the aspects that should be closely considered in the solution. Color influences the user experience since it can convey information (e.g. a scale of colors showing the temperature for a certain spot) and highlight information (e.g. a red warning). To guarantee consistency throughout the interface, a scheme of colors should be defined *a priori* [1].

According to Kenneth Moreland on his paper ”Diverging Color Maps for Scientific Visualization,” [8] most websites use a rainbow color map that uses all the most saturated colors. This color map faces different problems when users try to move from colors back to numbers since colors do not follow a natural order, the changes between colors are not uniform and there are colors sensitive to deficiencies in vision. The author proceeds to reasoning on how to choose a better color map. He mainly focuses on diverging color maps where they transition between color components by passing through an unsaturated color like yellow or white. Diverging color maps are mainly useful because they divide the values into three logical regions (low, medium and high values). The only problem with these color maps is the lack of natural ordering of colors which is solved by choosing two different colors that are mostly associated with ”low” and ”high” meanings. A good example is to use blue for low levels of temperature as it is often associated with ”low” and use red for high levels of temperature as it is often associated with ”warm”.

By looking at Figures 3.6(a) and 3.6(b) we can see two different websites using color to represent the different intensities of wind over the same area. They both use the blue color to represent lower levels of intensity and red to represent higher levels of intensity.

Glyphs. Another useful way to represent vector data is the use of glyphs. Glyph-based visualizations are often used when we need to represent multiple variables. Glyphs are essentially visual signs that can make use of visual features such as shape, color, texture, size, orientation, aspect ratio or cur-



(a) Wind intensity and direction over Portugal, source: winguru.cz

(b) Wind intensity and atmospheric pressure over Portugal, source: windy.com

Figure 3.6: Use of color, glyphs and isolines to represent data attributes.

vature to represent attributes of data. They are normally recognizable individually but can be spatially connected to convey topological relationships between data records [9].

As it is possible to see on Figure 3.6(a), two variables are represented at the same time. With color, it is possible to check the intensity of the wind over the areas represented and with glyphs, it is possible to see the direction of the wind. These glyphs are dots that represent the wind and the slight movement that is possible to check on the website indicates the direction of the wind, thus effectively conveying more information to the user.

Isolines. The last form to represent data is using isolines (or contour lines). These are essentially lines that connect different points on a map with the same value for the attribute we are representing. When these lines lie close together, they represent a steep slope while they represent a gradual scope when far apart.

On Figure 3.6(b) we once again use color to represent wind intensity but this time we use isolines to connect different regions that share the same values of atmospheric pressure.

3.4 Web Development Technologies

Three languages make up the core of the World Wide Web and thus are essential in the web development world. Hypertext Markup Language (HTML) is the language used to provide the content to be displayed by web browsers. Cascading Style Sheets (CSS) is the language used to change the presentation, formatting, and layout of web pages. Javascript (JS) is a scripting language used to control the behavior of the different elements on a web page and provide interactivity [10]. These three languages combined make up the foundation for the technologies we are going to study in the following subsections.

3.4.1 Responsive Design Frameworks

As previously seen, having a responsive web design is a major principle when we talk about web development. Nowadays, a large number of different screen sizes exist so designers need to figure out how to deliver the best experience across all those devices. Because of that, the number of HTML/CSS frameworks has increased substantially since they allow for faster results without having to add a lot of code [11].

These frameworks are packages that contain files with standardized code. The usual components are CSS code to create the layout grid, typography style definitions, fixes for cases of browser incompatibility and standard CSS classes to style the different components [12].

They address the majority of the existing screen sizes which is fundamental when talking about responsive design. Thus, we are going to compare some of those frameworks in order to analyze which is the best suited for this project. We chose to compare Bootstrap, Foundation, and Skeleton because they are open source and are the 3 most commonly used [13].

Bootstrap.[4] Is a powerful front-end framework developed at Twitter in 2010. The documentation is well structured and therefore it is a fairly easy framework to learn. It is the most popular framework at the moment, having 138k stars on GitHub, and thus it is possible to find help everywhere on the internet either through examples or asking questions in forums. It uses Syntactically Awesome Style Sheets (SASS) and Leaner Style Sheets (LESS) as preprocessors and weights around 8Mb.

Foundation.[5] Was created by ZURB in 2008. The documentation is excellent, especially because of the video tutorials that the website makes available. Although it is the second most popular framework it is significantly behind Bootstrap with around 30k stars on GitHub. It uses SASS as a pre-processor and the complete version weights around 1.6Mb.

[4] https://getbootstrap.com
[5] https://foundation.zurb.com

Skeleton.[6] Is a lightweight CSS framework created by Dave Gamache. The documentation is not extensive with just a few examples on the website. It is less popular than the previous two frameworks, having 17k stars on GitHub. Less experienced developers might get stuck when facing problems since there isn't much help. It uses both SASS and LESS as pre-processors and weights around 23Kb.

3.4.2 JavaScript Frameworks

As HTML and CSS are only capable of providing static content, JS is used to provide the interactivity web applications need. Nowadays, modern browsers have JS engines implemented to provide ways of manipulating the Document Object Model (DOM). This lets programmers to dynamically change the appearance of web pages at runtime [10].

The evolution of both browsers and JavaScript, lead to the emergence of JavaScript Frameworks (JSF). These frameworks are a set of functions and tools that were developed to facilitate the development of web applications across all devices. They allow for reusable and more organized code, consequently reducing the complexity of the development and saving time for the developers [14].

We are going to compare three of the most widely used JavaScript frameworks (Angular, React, and Vue) regarding their positive and negative aspects so we can make a decision on which framework is most suited for this work.

Angular.[7] An initial version called Angular JS was developed by Google in 2010 and was the first of the modern GUI frameworks. In 2016 Google released a completely new version which shifted the development language from JavaScript to TypeScript (TS). Although this shift benefited developers with other strongly-typed languages it significantly worsened the learning curve for beginners. It includes modern concepts such as component-based architecture, directives, improved data-binding, and dependency injection.

It is well suited for large-scale projects due to its high modularity. By using individual modules in the development it is possible to have isolated and independent code that allows having different teams working on the application independently.

Angular uses real DOM which is harder to handle and more prone to bugs than the virtual DOM. This also affects performance since whenever an element is updated the DOM needs to be rendered again.

Its documentation is not good for beginners since, in many of the guides, a very detailed view of the framework is presented. This leads beginners to waste time trying to find what they need. Nevertheless, it has a strong community backing it which results in constant updates and releases [10].

[6] http://getskeleton.com
[7] https://angular.io

React.[8] Is a JavaScript framework developed by Facebook in 2013. It follows a component-based architecture but functions like routing, data-binding or dependency injection require additional modules to be installed.

JSX is a syntax extension that allows using HTML elements as if they were JavaScript values. Although it is possible to use React without JSX it is not advisable to. It can be seen as a disadvantage because of the need to learn another syntax but it brings serious benefits like simpler and reusable code and protection from code injections.

It uses the virtual DOM which is a virtual representation of the real DOM. Whenever a component of the DOM changes state, a comparison is made between the current and previous versions and a calculation is performed to find the best way to make these changes. Only after that, the real DOM is updated. Overall, this contributes to better performance.

It provides nice documentation with several useful guides and structured API documentation. It is beneficial not only for beginners but also for experienced developers since this framework has an immense focus on the development experience. It has a huge community behind it (especially because of Facebook) making it easy to find solutions for problems on the internet [10].

Vue.[9] Is a progressive frontend framework developed by Evan You in 2014. In terms of basic structure and templating it is similar to React since both use a component-based architecture and keep the focus on the core library. In order to have functions like routing or data-binding, other libraries are required.

It has removed some of the drawbacks of the previous two solutions and provides good flexibility and customizability. It is the smallest of the three in terms of weight. It also uses a virtual DOM which helps it have a better performance when updating the state of the interface.

It has friendly documentation especially for beginners with lots of examples and explanations. In comparison with the other two frameworks, it is possible to note that its community is way smaller mostly due to the other two having big companies backing them while Vue is an independent project [10].

3.4.3 Server Side Development

As we have seen previously, the client-side code mainly handles the structure and presentation of information to the users. But, we must control what kind of information is being sent to the user. This is why we need server-side code. This code which runs on a web server is responsible for retrieving data from a database and dynamically display it on the website. Server-side programming allows the creation of

[8] https://reactjs.org
[9] https://vuejs.org

personalized user experience since it is possible to highlight or save content that is more relevant to each user depending on his/her habits. For example, saving the user's most searched beaches. Therefore, we are now going to compare two different technologies that are being used for server-side coding, to choose what is the best one for this work.

PHP.[10] PHP (Hypertext Preprocessor) was developed in 1994 by Rasmus Lerdorf and is an open-source server-side scripting language.

Mostly suited for CPU intensive applications and where there is not much interaction between client and server. It is supported by many hosting services which makes the deployment and integration easier than with NodeJS. It has the serious disadvantage of being synchronous which affects the performance of applications. PHP is mostly used with traditional SQL databases, but it is difficult to implement the connection with NoSQL databases [11].

NodeJS.[12] NodeJS was developed in 2009 by Ryan Dahl and is an open-source JavaScript run-time environment to execute JavaScript code on the server-side.

Since it is built on JavaScript, developers do not need to switch languages since both server-side and client-side use JavaScript. Thus, it is easier to learn and start working with since developers just need to use JavaScript. When compared to PHP, NodeJS is a lot faster due to being built on a V8 engine, its continued server connection, and its callback functions that process multiple requests at the same time. It has included the Node Package Manager (npm) which offers a set of available reusable components. These components can be easily installed via an online repository. It works perfectly with NoSQL databases and has various libraries to work with SQL [13].

3.4.4 Database

In order to avoid too many calculations and to increase the performance of the system it is important to have a database where we can store information. To provide a good user experience, the time between the user request and the response must be close to minimum and for that a database that offers a good performance must be chosen. At first there is the need to choose between the two types of databases, Relational or NoSQL and then we can move to decide which is the best one [15].

Relational Databases. These type of databases are most efficient when we are dealing with a specific data structure, where we know at design-time which is the type of data we are dealing with. The

[10] https://www.php.net/
[11] https://hackernoon.com/nodejs-vs-php-which-is-better-for-your-web-development-he7oa24wp
[12] https://nodejs.org/
[13] https://www.geeksforgeeks.org/php-vs-node-js/

data is stored in separate columns which makes the insert and read operations slower. One advantage of Relational databases is that it is easy to manipulate the data with queries [16].

NoSQL Databases. NoSQL databases are based on key-value structures allowing the data to scale horizontally over time. Although scaling horizontally helps in terms of performance, the trade-off is that it looses the consistency of their data. Since the data is stored in JavaScript Object Notation (JSON) format, NoSQL databases have high rates of insert and read operations [16].

3.4.5 Web Map Libraries

In this application, there is a need to have data represented on a map. More specifically it needs an interactive map of Portugal that allows users to select specific locations. To avoid unnecessary work on developing an interactive map from scratch, many different JavaScript libraries can be used to develop a map that specifically suits website's needs. In this section, we compare and analyze several existing libraries to find the one that is most suitable for this application.

Leaflet.[14] Is one of the most adopted open-source JavaScript libraries for interactive maps. The basic library is relatively small in size making it a great option for applications in which load time needs to be small. On the other hand, it may lack some of the necessary functionalities that need to be complemented with the use of plugins. Due to its large community, there are hundreds of plugins that can satisfy the different needs of the developers by adding new functionalities. It is well documented but lacks concrete implementation examples making it hard for beginners. Once the developer has more experience it allows for faster development because it only needs a few lines of code to implement its features. Overall it is used whenever we are not building a complex GIS application but instead a simple visualization of a map.

Mapbox GL.[15] Is another open-source JavaScript library used to render interactive maps. It was built by experienced developers making it an advanced mapping solution. It provides a wide range of tools that allow users to have full control over map styling. This library has detailed documentation with several real examples that are helpful for beginners. A major drawback of this solution is that it starts to have a cost associated with 50,000 monthly loads. Although it won't be a problem in the short term, in a medium to long term it is not affordable having to pay at least $250 per month.

OpenLayers.[16] Is another open-source JavaScript library for displaying map data in web browsers.

[14] https://leafletjs.com
[15] https://docs.mapbox.com/mapbox-gl-js/overview/
[16] https://openlayers.org

It is possible to choose the map layer source using a tiled layer or vector layer from several map services. The product comes mobile-ready out of the box, compatible across every modern web browser and mobile devices. The documentation is nicely structured, but since it is very large, it may be difficult at the beginning. It has plenty of examples, but due to several versions released some of the examples that can be found online are outdated. A big advantage is that it is possible to build custom lightweight profiles with just the components needed. Also, the core package comes with all the required functionalities that are essential for more complicated types of visualizations.

4

Solution

Contents

Now that we have established a solid foundation for every decision we made throughout this work, we are in a good position to explain the details of the work performed. Therefore, in this chapter we describe how the system is organized, by analyzing first the architecture of our solution and then each of the main components that make it work.

4.1 Architecture

It is first important to describe the architecture of the project in order to understand how the entire system works together to produce the final result, which is the visualization of surf forecasts. By looking at Figure 4.1, we can understand that there are 5 components that make up this project. On the left side and with the connections represented as dashed lines we see the 2 components that are external to us and therefore the only thing that was developed by us was the connection to their APIs to retrieve data. On the center and right side we have the 3 components developed by us which are the server, the database and the client application.

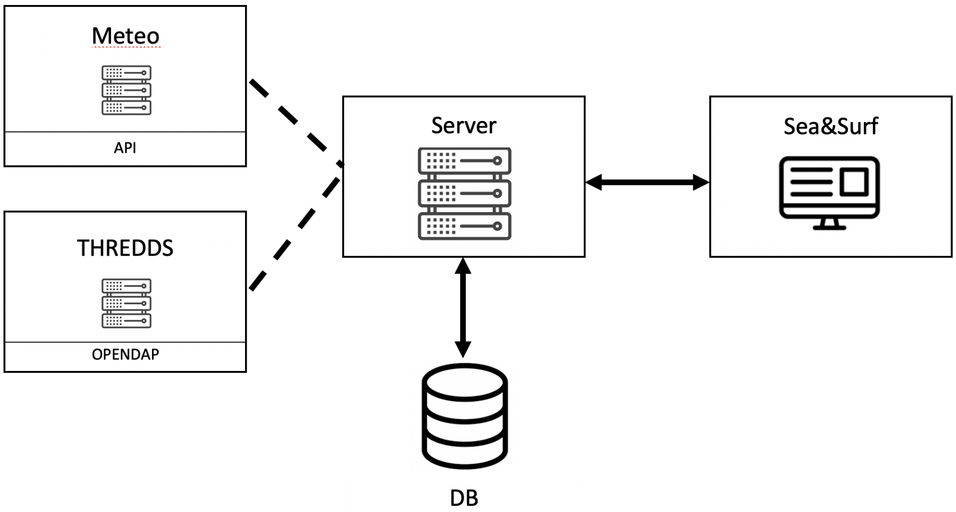


Figure 4.1: System architecture

As we previously mentioned, we took advantage of both weather and maritime forecasted data that Maretec produces. Both these data are coming from the external components previously mentioned, namely the weather data is retrieved from the meteo—Técnico server, and the maritime data from the Maretec THREDDS server.

The Sea&Surf component is the client-side of our application, which in this case will be a single-page application (SPA). This means that the website will consist of a single page that will be dynamically rendered based on user interaction.

The flow of data is as follows: the client requests information for a specific beach in Portugal to the server, which will check if it already has the data stored in the database. If the data is already stored on the database, the server can immediately return it to the client. If it does not have the data it will send requests to both Meteo and THREDDS servers to get the data needed and after receiving it, the server stores it in the database and sends it back to the client. In the next subsections we will look into more detail each of the components of the system architecture.

The last aspect that is important to note is the hosting. This project will be hosted in a server owned by the project sponsor (Maretec) and therefore there will be no costs associated with this project. The application will be available at `https://hawking.tecnico.ulisboa.pt/surf/` and the 3 components (database, server and client) will be running on this machine.

4.2 Database

In order to store the data collected and calculated by this solution, we decided to use a NoSQL database because as seen in 3.4.4 they offer better performance when compared to Relational databases which translates to faster response times between user requests.

In the category of NoSQL databases we chose the document-oriented database MongoDB[1]. This decision was based on the fact that it is one of the most popular NoSQL databases at the moment which helps in terms of development. Another point in favor is that it is a database already in use by the project sponsor (Maretec) so we avoided the installation part, and kept the consistency between projects.

By allowing to store data as a series of JSON documents, MongoDB gives our solution the perfomance needed for this kind of applications where users demand fast access to the data. JSON provides an easier and faster storage and retrieval of data records which makes the communication between the database and our server faster and easier to develop.

For the purpose of this project a schema name 'seasurf' was created in order to define the 2 Collections needed. The first Collection is called 'praias' and is used to define all beaches that can be searchable in the application. As we can see in Figure 4.2, we have a Document defined for each available beach with the following information:

- **Id** - Unique identifier for each Document
- **Name** - Name of the beach (E.g. ”Praia do Guincho”)
- **Orientation** - Orientation of the beach (E.g. ”N-S”)
- **Latitude** - Latitude of the beach (E.g. ”38.7325”)
- **Longitude** - Longitude of the beach (E.g. ”-9.4725”)

[1] https://www.mongodb.com

- **PositionThreddsMohid** - Array with the position in terms of row-column for the Mohid system (E.g. [72, 51])
- **PositionThreddsWaves** - Array with the position in terms of row-column for the Waves system (E.g. [63, 44])

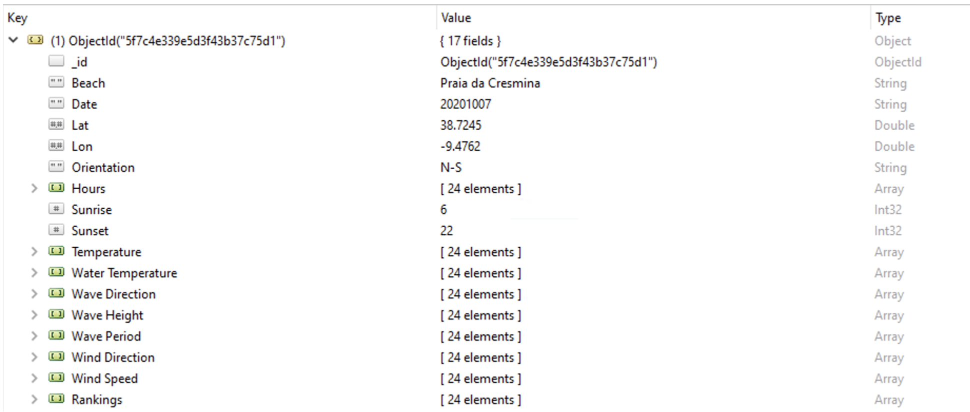
Key	Value	Type
(1) ObjectId("5f21f5b9b21fc7750425cbb7")	{ 8 fields }	Object
_id	ObjectId("5f21f5b9b21fc7750425cbb7")	ObjectId
Name	Praia do Guincho	String
Orientation	N-S	String
Position	{ 2 fields }	Object
Latitude	38.7325	Double
Longitude	-9.4725	Double
PositionThreddsMohid	[2 elements]	Array
PositionThreddsWaves	[2 elements]	Array

Figure 4.2: Example of a Document inside 'praias' Collection

The second Collection is called 'data' and stores all information regarding each beach for each specific day. As we can see in Figure 4.3, there is a Document defined for each available beach with the following information:

- **Id** - Unique identifier for each Document
- **Beach** - Name of the beach (E.g. "Praia da Cresmina")
- **Date** - Corresponding date for this information (E.g. "20201007")
- **Lat** - Latitude of the beach (E.g. "38.7245")
- **Lon** - Longitude of the beach (E.g. "-9.4762")
- **Orientation** - Orientation of the beach (E.g. "N-S")
- **Hours** - Array with hours the system has information for (E.g. [0,1,...,23])
- **Sunrise** - Hour for the Sunrise (E.g. 6)
- **Sunset** - Hour for the Sunset (E.g. 22)
- **Temperature** - Array with Temperatures for each hour in Celsius (E.g. [17.5,18.0,...,16.5])
- **Water Temperature** - Array with Water Temperatures for each hour in Celsius (E.g. [15.4,13.5,...,12.6])
- **Wave Direction** - Array with Wave Directions for each hour in degrees (E.g. [323,322,...,315])
- **Wave Height** - Array with Wave Height for each hour in meters (E.g. [1.75,1.80,...,1.60])
- **Wave Period** - Array with Wave Period for each hour in seconds (E.g. [7.75,7.80,...,7.60])

- **Wind Direction** - Array with Wind Directions for each hour in degrees (E.g. [323,322,...,315])
- **Wind Speed** - Array with Wind Speed for each hour in m/s (E.g. [10.1,11,...,9])
- **Rankings** - Array with Rankings for Surf for each hour (E.g. [2.9,2.7,...,1.4])



Key	Value	Type
(1) ObjectId("5f7c4e339e5d3f43b37c75d1")	{ 17 fields }	Object
_id	ObjectId("5f7c4e339e5d3f43b37c75d1")	ObjectId
Beach	Praia da Cresmina	String
Date	20201007	String
Lat	38.7245	Double
Lon	-9.4762	Double
Orientation	N-S	String
Hours	[24 elements]	Array
Sunrise	6	Int32
Sunset	22	Int32
Temperature	[24 elements]	Array
Water Temperature	[24 elements]	Array
Wave Direction	[24 elements]	Array
Wave Height	[24 elements]	Array
Wave Period	[24 elements]	Array
Wind Direction	[24 elements]	Array
Wind Speed	[24 elements]	Array
Rankings	[24 elements]	Array

Figure 4.3: Example of a Document inside 'data' Collection

4.3 Server

In order for our web application to have the data needed to produce the visualizations we created a server using NodeJS. The decision to use NodeJS comes from the study in Chapter 3.4 where we saw that since we are developing a highly interactive system where users can constantly request new information from the server we needed to have a faster engine that processes multiple requests at the same time. Also, since NodeJS is built on JavaScript, we did not need to learn a new language, saving us time, and making it easier to change between client-side to server-side development.

This server performs 4 main tasks: extract data from forecast models; connection to the database; calculate surf rankings; provide APIs.

4.3.1 Extract data from forecast models

As said in a previous section, we need the data produced by Maretec in order to create a website that shows surf forecasts. Maretec uses a Thematic Realtime Environmental Distributed Data Services (THREDDS) Data Server which is a web server that allows to store data from scientific datasets to be accessed by anyone. We already saw in Table 3.1 the 7 different variables we needed for this project which are the air temperature, wind speed, wind direction, water temperature, wave height, wave period and wave direction.

The first 3 variables correspond to the meteorological data and can be accessed and retrieved via an API provided by meteo—Técnico. This API facilitates our work in the sense that we do not need to access the files stored on any THREDDS Data Server and read them in order to have the information we need. Instead we can simply connect to an Application Programming Interface (API) via Hypertext Transfer Protocol (HTTP) GET requests to the following url: `http://meteo2-ciist.tecnico.ulisboa.pt:8080/api`. In Listing 4.1 we can see the code for the request to this API. For the request to be successful we must send our authentication credentials (user and password), we must provide the coordinates of the beach we intend to retrieve information and the variables we are trying to retrieve. It is important to mention that for this API the longitude comes first on the request followed by the latitude (which is the opposite of the normal coordinates convention). The variable names t2, ws10 and wd10 correspond to the air temperature, wind speed and wind direction respectively.

Listing 4.1: Request to the Meteo—Técnico API

```

//Meteo API uses longitude in first place and latitude in second
request('http://meteo2-ciist.tecnico.ulisboa.pt:8080/api/wrfzip?coord=${lng
    },${lat}&vars=t2,ws10,wd10', {
        'auth': {
            'user': 'XXXXX',
            'pass': 'XXXXX'
        },
}
```

In what regards the hydrographic data, the approach is way more complex since we do not have access to any API to retrieve the data. Therefore we must access the files in the THREDDS server and retrieve the information we need. This THREDDS server stores the data produced by the forecast models in Network Common Data Form (NetCDF) which is suitable for scientific data to be stored in arrays of dimensions. We can then access any of this dimensions such as latitude, longitude or time to retrieve the data we are interested on.

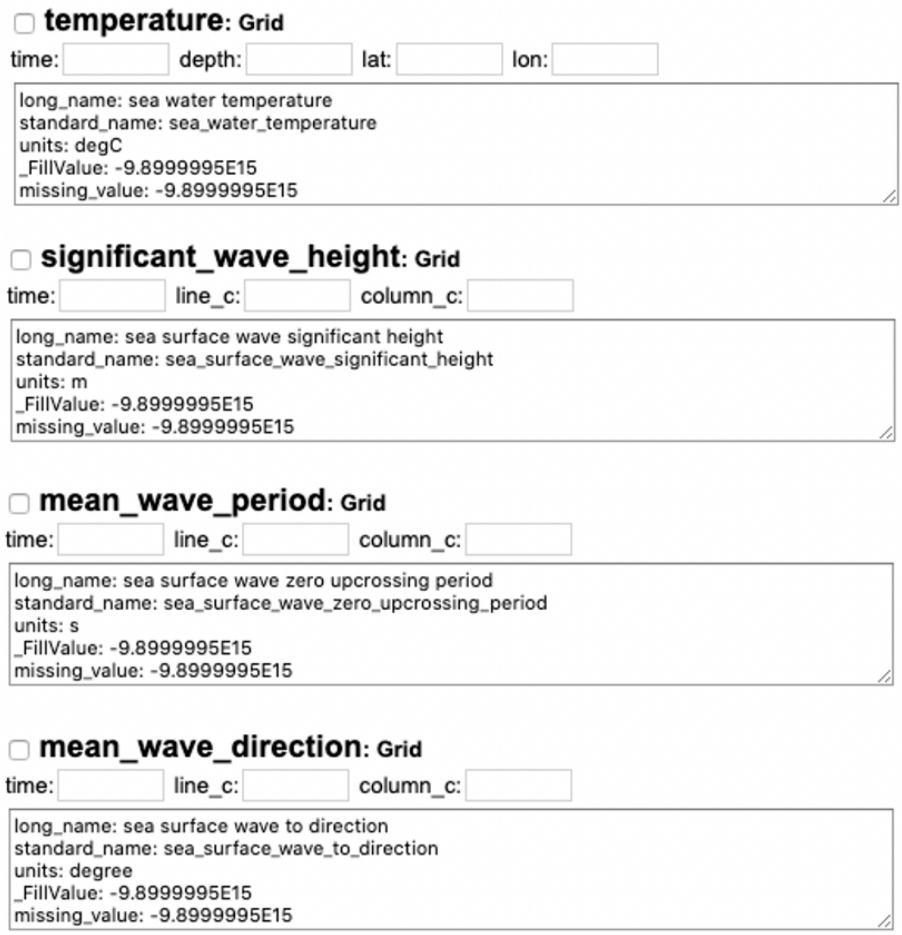
In order to query a subset of a NetCDF file that is stored on a remote server we must implement the Open-source Project for a Network Data Access Protocol (OPENDAP) with a Python library called Pydap[2]. This library allows us to work with the different data dimensions on-the-fly saving us bandwith and time.

The four remaining variables we need are produced by two different forecast models which means we need to read from two different files. While water temperature is forecasted by the MOHID-WATER

[2] https://pypi.org/project/Pydap/3.0/

model, wave height, wave period and wave direction are forecasted by the WW3-WAVES model.

In these two files each of the 4 variables we are trying to retrieve are represented in the form of a grid with several dimensions. From what we can see in Figure 4.4 the water temperature has 4 dimensions which are time, depth, latitude and longitude while the other three variables have 3 dimensions which are time, line-c and column-c. The time dimension in each of them is an array of the 24 hours of the day, and since we need the information for all hours, we are always going to specify the entire dimension when retrieving the data. For the depth dimension we are going to specify the value in the dataset that is closest to 0, since we are looking for the temperature of the water at the sea level where surfers surf waves. Regarding the latitude and longitude for the water temperature and line-c and column-c for the other variables they mean exactly the same thing, which are the coordinates of the spot we intend to get the data from.



temperature: Grid
time: depth: lat: lon:

long_name: sea water temperature
standard_name: sea_water_temperature
units: degC
_FillValue: -9.8999995E15
missing_value: -9.8999995E15

significant_wave_height: Grid
time: line_c: column_c:

long_name: sea surface wave significant height
standard_name: sea_surface_wave_significant_height
units: m
_FillValue: -9.8999995E15
missing_value: -9.8999995E15

mean_wave_period: Grid
time: line_c: column_c:

long_name: sea surface wave zero upcrossing period
standard_name: sea_surface_wave_zero_upcrossing_period
units: s
_FillValue: -9.8999995E15
missing_value: -9.8999995E15

mean_wave_direction: Grid
time: line_c: column_c:

long_name: sea surface wave to direction
standard_name: sea_surface_wave_to_direction
units: degree
_FillValue: -9.8999995E15
missing_value: -9.8999995E15

Figure 4.4: OPeNDAP Dataset Access Form for the hydrographic variables

Since we are talking about dimensions which are arrays, whenever we need to find values for a specific location we must find which positions of each array correspond to each coordinate. This means that we must first find which are the x and y positions in each array that correspond to the latitude and longitude of each beach.

For this project we are considering that these files are not going to change very often, so we can say that we will only need to calculate the positions of the arrays that match the coordinates once for each beach. Whenever we add a new beach location to our system, it will calculate the x and y positions of the beach for each of models (MOHID-WATER and WW3-WAVES) and store them on the database together with the information of the beach. This means that every time a user requests information for a beach we already have the positions stored on the database and it will be faster to retrieve the results since we do not need to calculate those positions again.

One problem we encountered when we started to calculate the positions was that these models do not have a continuous set of points represented on the dataset which means that we could not look for the exact coordinates of the beach but instead look for the closest ones available on the dataset. After that we found that sometimes the closest point to the one we are requesting can be a point in land. This was another problem since if we try to use that point we would get infinity for each of the values we are looking for because there can't be valid values for the hydrographic data on land.

Algorithm 4.1: Closest Water Point on the Dataset

```
begin
    infinite ⟵ −98999995
    array4Points ⟵ getFourClosestPoints()
    for Point in array4Points do
        if value >= infinite then
            savePoint(Point)
    if savedPointsLength > 1 then
        CalculateSmallestDistance(savedPoints)
    else if savedPointsLength == 0 then
        array12Points ⟵ getTwelveClosestPoints()
        for Point in array12Points do
            if value >= infinite then
                savePoint(Point)
        CalculateSmallestDistance(savedPoints)
```

Based on these two problems we developed an algorithm to find the closest water point of each beach in each of the datasets. In Algorithm 4.1 we explain that first we are getting the four closest points on the grid and store all that are returning valid values ie, that are water points. If we have more than one water point we calculate the one that is closer to our original point. Otherwise we will get the twelve outer points and repeat the procedure. Once we are done with this algorithm we are supposed to get the closest water point and its corresponding x and y position to be stored on our database for further usage.

As we said previously, this operation only needs to be performed once for each beach, and we are ready to start retrieving the information we need from the models. In Listing 4.2 we can see how we are extracting the data from the WAVES dataset. First we load the dataset information regarding the three variables (wave period, wave height and wave direction). Second we provide each subset of the data the time, x position and y position dimensions to get the correct data. Finally we format the extracted data to be sent in the JSON format.

Listing 4.2: Python extraction of data from the WAVES model

```python
def extractWavesDataset(dataset, wavesPos):
    meanWavePeriod = dataset['mean_wave_period']
    significantWaveHeight = dataset['significant_wave_height']
    meanWaveDirection = dataset['mean_wave_direction']

    coordX = int(wavesPos[0])
    coordY = int(wavesPos[1])

    #Retrieve the data we need
    grid = meanWavePeriod.array[:,coordX,coordY]
    grid.reshape(1,25)
    meanWavePeriodList = grid.data[0].tolist()

    grid = significantWaveHeight.array[:,coordX,coordY]
    grid.reshape(1,25)
    significantWaveHeightList = grid.data[0].tolist()

    grid = meanWaveDirection.array[:,coordX,coordY]
    grid.reshape(1,25)
    meanWaveDirectionList = grid.data[0].tolist()

    jsonData = { 'Wave Period': meanWavePeriodList, 'Wave Height':
        significantWaveHeightList, 'Wave Direction': meanWaveDirectionList}

    return jsonData
```

4.3.2 Database Update

Our server must be connected to the database so we can store the information we are retrieving from the different models. As previously mentioned whenever a user requests information from a specific beach, our server first checks if the data is already stored on the database and can be thus immediately returned. If the data is not stored yet, the process described in the previous subsections is done and the output data is stored on the database. By storing this information we are improving the system performance since we do not need to make complex calculations and access the different models' dimensions once again.

Besides this flow of user requesting the data, our system has a task scheduled to run 4 times a day (at 12am, 4am, 12pm, 20pm) in order to check for new information available as well as to update existing information that was recalculated by the forecast models. This task retrieves information from models not only regarding the current day but also the next two days. With this we are ensuring that the data displayed in our visualization is always up to date.

Another task is also scheduled to run 4 times a day (at 1am, 5am, 13pm, 21pm), 1 hour after the first task in order to calculate the surf rankings based on the data already retrieved.

4.3.3 Calculation of Surf Rankings

In order to take advantage of the collected data and to make this project to have something that really distinguishes it from the other solutions on the market, we decided that one of the goals was to develop a surf ranking.

From the analysis in Chapter 2 we were able to understand the different components of surf and which variables affect the quality of a beach for surfing. By considering this analysis and the corresponding collected data we were able to develop an algorithm capable of defining for each beach and for each day a ranking from 0 to 5, where 0 means the beach is not good for surfing and 5 means that the beach is excellent for surfing.

With this ranking we are able to give the users more information regarding the beaches for surfing purposes. Surfers can look at this ranking and realize faster whether it is good or not to go surfing without having to look at all the variables. This ranking is also helpful when users intend to go surfing to a beach they normally do not go and therefore do not know which are the best conditions for that beach. Another positive aspect is the fact that not all surfers know how to interpret a surf forecast and therefore need extra help to understand how the beach is for surfing. Although there are lots of different types of surfers we developed a standardized ranking that can be used by any surfer.

In Algorithm 4.2 we have the core functions executed in order to calculate the surf ranking. This algorithm takes advantage of the data retrieved from the models and already stored on the database to

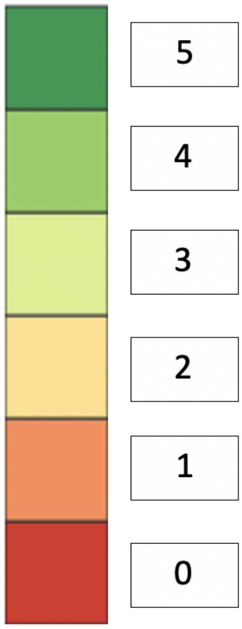


Figure 4.5: Surf Ranking Color Scale

calculate individual rankings for each of the variables. As shown in Chapter 2 the main variables that allow us to determine if a beach has favorable conditions are wind speed, wind direction, wave direction, wave height and wave period. Beach orientation is an important variable to determine if the wind and wave directions are favorable for the beach as we will explain in more detail later in this section.

Thus our algorithm is divided in 5 functions (one for each of the main variables) where we perform individual calculations of each variable. Each of the functions will return a value from 0 to 1, where 0 corresponds to the fact that the variable values are considered to be bad, while 1 corresponds to good variable values. After those calculations are done we sum all the results to get our final ranking for each hour. We are going to detail the functioning of each of the 5 functions.

Wind Direction Ranking Calculation. For this calculation we need the value of the wind direction in degrees for each hour and the beach orientation.

We have previously seen that offshore winds (coast out to the sea) are the best ones for surfing and onshore winds make it harder to surf the waves. Also, wind directions that are parallel to the beach orientation are not good because they break the waves flow.

In Figure 4.6 we can see an example for Praia do Guincho. For a beach with such orientation, favorable wind directions are those that have values comprehended in the green box because they are coming from the cost towards the sea. So for any values comprehended in the offshore winds we return the value of 1, while for any other wind directions we return the value 0.

Algorithm 4.2: Calculation of Surf Ranking

```
begin
    beachOrientation ⟵ beach['Orientation']
    windDirection ⟵ beach['WindDirection']
    windSpeed ⟵ beach['WindSpeed']
    waveDirection ⟵ beach['WaveDirection']
    waveHeight ⟵ beach['WaveHeight']
    wavePeriod ⟵ beach['WavePeriod']
    for hour in arrayOfHours do
        if windDirection! == undefined then
            checkWindDirection(beachOrientation, windDirection[hour])
        if windSpeed! == undefined then
            checkWindSpeed(windSpeed[hour])
        if waveDirection! == undefined then
            checkWaveDirection(beachOrientation, waveDirection[hour])
        if waveHeight! == undefined then
            checkWaveHeight(waveHeight[hour])
        if wavePeriod! == undefined then
            checkWavePeriod(wavePeriod[hour])
        FinalRanking = windDirectionRank + windSpeedRank + waveDirectionRank +
          waveHeightRank + wavePeriodRank
```

Wind Speed Ranking Calculation. For this calculation we only need the value of the wind speed in m/s for each hour for the requested beach. There are two main aspects we need to take in consideration for this calculation. First, and as we have seen earlier in this document most of the times high winds are not good for surf because they make the sea very unstable.

In order to establish a scale for wind speeds we used the Beaufort scale [3] which defines 12 levels for the categorization of the correspondence between sea conditions and wind speed. The last 3 levels consider wind speeds higher than 25m/s which correspond to extremely adverse conditions such as storms and hurricanes. Since when there is extremely adverse conditions surfers do not go into the water we decided that our scale for wind speed would go from 0m/s to 25m/s.

With these two aspects in mind any wind speed values close to 0m/s will give a score of 1, while wind speed values close to 25m/s will give a score of 0 in this calculation.

Wave Direction Ranking Calculation. For this calculation we need the value of the wave direction in degrees for each hour and the beach orientation.

We have previously seen that for the wave direction we apply the opposite reasoning of the wind direction. So, wave directions that come from the sea towards the coast are the best for surfing.

In Figure 4.7 we can see an example for Praia do Guincho. For a beach with such orientation,

[3] https://en.wikipedia.org/wiki/$Beaufort_scale$

Figure 4.6: Favorable Wind Directions for Praia do Guincho

favorable wave directions are those that have values comprehended in the green box because they are coming from the sea towards the cost. So for any values of wave directions from the sea to the coast we return the value of 1, while for any other wave directions we return the value 0.

Wave Height Ranking Calculation. For this calculation we only need the value of the wave height in meters for each hour for the requested beach. There are two main aspects we need to take in consideration for this calculation. First, and as we have seen earlier in this document, for most surfers the bigger the wave the better it is.

Second, in order to establish a scale for wind speeds we used the Douglas sea scale [4] which defines 9 levels for the categorization of the correspondence between sea conditions and wave height. These levels have values that go from 0 meters to 14+ meters.

With these two aspects in mind any wave height values close to 0 will give a score of 0, while wave height values close to 14 meters or higher will give a score of 1 in this calculation.

Wave Period Ranking Calculation. For this calculation we only need the value of the wave period in seconds for each hour for the requested beach. There are two main aspects we need to take in consideration for this calculation. First, and as we have seen earlier in this document the bigger the wave period the bigger and longer the waves are.

[4] $https://en.wikipedia.org/wiki/Douglas_sea_scale$

Figure 4.7: Favorable Wave Directions for Praia do Guincho

Second, in order to establish a scale for wind speeds we used the information available at Surf Today[5] which defines 5 different levels for wave periods.

With these two aspects in mind we considered that for wave periods less than 6 seconds a score of 0 would be returned, for wave periods between 6 and 8 a score of 0.25, for wave periods between 8 and 10 a score of 0.5, for wave periods between 10 and 12 a score of 0.75, and for wave periods higher than 12 a score of 1 would be given.

4.3.4 Provide APIs

The server is also responsible for providing APIs in order for our Web application to make requests and get the data to be displayed. We have developed 3 different APIs that are responsible for receiving HTTP GET requests from the client and returning the data that was requested, which are: the beachAPI, the dataAPI and the rankingsAPI.

The beachAPI is the one responsible for retrieving all the beaches' information available in our application. It will receive a parameter which is the input string corresponding to the beach name that the user intends to get information. Based on this string, the server will retrieve all beaches from the collection 'praias' which name contains the specified string. The information retrieved by the server to the client is the one previously shown in Figure 4.2.

[5] https://www.surfertoday.com/surfing/the-importance-of-swell-period-in-surfing

The dataAPI is the most complex one and is responsible for retrieving all the data (both weather and maritime) regarding a beach for a specific day. Therefore this API needs to receive the date and the beach name and coordinates as parameters. The server will then search for records for the specified date and beach on the collection 'data'. If it finds a matching record it will send immediately the data back to the client. If it does not find any matching record it will start the process explained in 4.3.1 in order to extract the data from the forecast models.

The rankingsAPI is the one responsible for retrieving surf rankings for all the locations defined in our application. This API needs one request parameter which is the date for which the user wants to see the surf rankings information. The server will then search for all the records that match the requested date on the collection 'data' and send that information back to the client. It is important to note that although the surf rankings are defined as an array of dimension 24 because of the 24 hours of the day, we are only sending the ranking value for the current hour of the day, as we can only represent one beach on the map at a time (we will talk about this feature in the next subsection).

4.4 Web Application

The Web Application is the last of the 3 components we have developed and it is represented as 'Sea&Surf' in Figure 4.1. As pointed out in the beginning of this chapter this is the client-side of our system and is a single-page application.

By creating a Single Page Application (SPA) we are improving our users' experience since we do not need to refresh the entire interface but only the necessary components. With this approach we load the entire application to the client at once and whenever the client needs new information it will use the APIs previously mentioned to communicate with the server [17].

As far as technologies chosen for this component, we used HTML and CSS to provide the content and change the layout of the website. Like any other technology, these two programming languages are constantly evolving and introducing new features, therefore we used their latest versions, which are HTML5 and CSS3. Regarding Javascript frameworks, we decided to first exclude Angular from the list for several reasons. It is more complex than the other two frameworks, has a steeping learning curve since the documentation is not well suited for beginners, and has a worse performance mainly because it uses the real DOM to update elements on a webpage. Between React and Vue, we decided to go with React mainly because it has a bigger community behind it and extensive documentation.

Figure 4.8 shows the interface developed and its 10 components represented with red rectangles and numbers.

The following list describe each of the 10 component and its functionalities.

- **1** - **Logo**. This is the logo that represents the entire developed system with reference to Maretec,

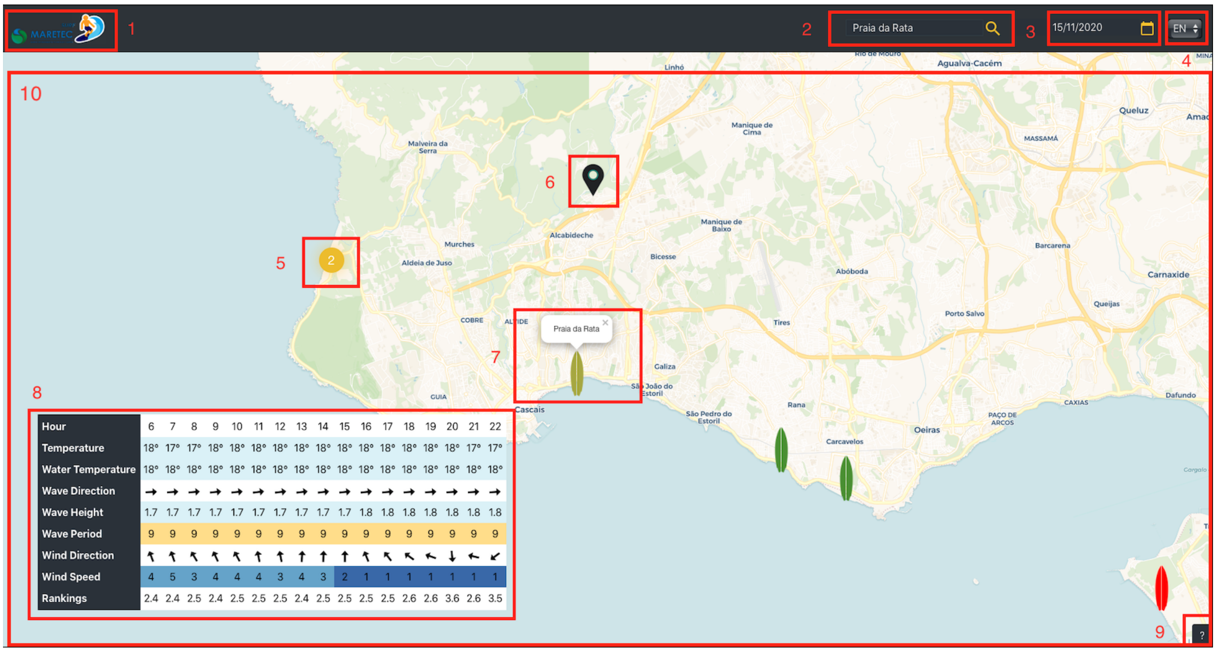


Figure 4.8: Interface Components

the parent organization of this project.

- **2** - **Search Bar**. Within the search bar, users can type any text in order to search for the beach they need to get information from. Whenever a user types in the search bar a new request is being made to the beachAPI, and several rows of suggestions are presented to the user in order to facilitate navigation. Whenever the user clicks on one of the suggestion rows some triggers to other components are performed.

- **3** - **Calendar**. This allows users to change the selected date in order to get information for the requested beach for the future or the past. With regards to the future users might need to know how are the surf conditions for the next 3 days so they can plan if its worth to go surfing. With regards to the past users might want to check how were the surfing conditions in the previous days for teaching or prediction purposes.

- **4** - **Language Switcher**. As we might have surfers from different nationalities surfing the Portuguese waves it is important that those who do not speak Portuguese can still be able to access this website and understand the information displayed by seeing it in English.

- **5** - **Cluster of Beaches**. Whenever there are multiple beaches close to each other, it is important to cluster them so that the visualization does not get too crowded and therefore impossible to interpret. Whenever a user sees a yellow circle with a number, he/she is informed that there are those number of beaches close to each other in that specific location.

- **6** - **User's Location**. This represents an approximate user's location which helps the system to identify and zoom-in in that area so that users can immediately see the available beaches around them.
- **7** - **Beach**. Every beach available in our system is represented with a surfboard icon and a tooltip indicating the name of that beach. Furthermore, the surfboard icons have a color associated based on the surf ranking calculated for each beach. The colors and corresponding ranking values can be seen in Figure 4.5. In order for this surfboard icons to show up, a request is made to the rankingsAPI.
- **8** - **Data Table**. This is where both weather and maritime information needed for the practice of surf is available. This table is available for a specific beach and day at a time. There are two ways to trigger the information table to show up: first is by clicking on a suggestion item on the search bar and second is by clicking on a surfboard icon. In order to populate the table with data a request is made to the dataAPI.
- **9** - **Help**. Since users are not familiarized with the correspondence between the colors and the values of the surf rankings, they can use this help button to understand that correspondence (popup with Figure 4.5 will show up in the middle of the screen).
- **10** - **Map**. The map serves as baseline for almost all other components, since it provides a representation of Portugal and its coast to the users. It is easier for users to see the locations represented on a map whenever they are searching beaches in a specific area. In order to implement the map on our project we chose the Leaflet web map library because it is a lightweight library that uses a mobile-first approach and that can be implemented with just a few lines of code.

One important aspect of the client-side application is the responsive design. As mentioned earlier in section 3.1.1, two techniques can be used to build a responsive website which are graceful degradation and progressive enhancement. Since the main goal of this work was to develop a new website, and since we started from zero, we used the second technique.

Using this technique meant that the focus started on designing for the smallest device and showing the most important information first and gradually evolve to larger screens where secondary information was also available. A core concept of the chosen technique is that developers can create a default, content-only page first and apply the layout afterward depending on the end-user device's size [18].

When applying the layout to the website, several aspects regarding responsive design techniques were addressed. Firstly, the meta tag that was introduced in the latest version of HTML (HTML5), which allows developers to take control of the viewport. According to W3C[6], the following line of code should

[6]https://www.w3.org

be present on all our website pages since it gives instructions to the browser on how to control the page's dimensions and scaling.

```
<meta name="viewport" content="width=device-width,initial-scale=1.0">
```

With this tag, we set the width of the web page according to the device's size so that the page automatically adjusts to the viewing screen.

The second aspect is that we designed our web pages based on a grid layout, which means that the pages are divided into columns to make it easier to place elements wherever we want. This layout uses percentages to scale elements accordingly to each other depending on the resolution of the screen [4].

Thirdly, in media objects like images and videos, we defined its max-width property to scale down images if needed and to guarantee images never get larger than their original size.

Another aspect is the media query which is one of the most useful modules of the latest version of CSS (CSS3). Media queries allow us to show and organize the different elements of a webpage within a single CSS document, depending on the size of the end-users browser window [18]. The next line of CSS is an example of a query that will be executed for a device size that is below 768 pixels.

```
@media only screen and (max-width: 768px) { /* CSS code */ }
```

The use of a framework will help us to address the majority of screen sizes and to accelerate the process of building different layouts for a variety of screen dimensions. For that reason we chose to use Bootstrap. Although it is heavier when compared to the other two frameworks (Bootstrap - 8Mb, Foundation - 1.6Mb, Skeleton - 23Kb), the difference is not significant and therefore can be ignored. The major advantages of Bootstrap are its popularity and documentation. These factors made it an easy framework to start with and to find solutions whenever we faced problems during development.

Another important aspect of the interface is the variables representation. As we have seen in Section 3.3, with the analysis of the forecast websites and the interviews with users, we obtained the attributes we must show (See Table 3.1) in the table of information. But to improve the user experience we defined a scale of colors to better help users to map the values of each variable into what exactly they mean (for example in Figure 4.8 we see that the value 9 for the wave period means this is a high value since the color is orange). With the help of ColorBrewer[7], we have defined a sequential color scheme to be used. This color scheme is colorblind safe, which takes care of one of the problems found when we performed the user's characterization.

On Figure 4.9 it is possible to see that the color scale used ranges from dark blue for smaller values of each variable to dark red for higher values of each variable.

[7] www.http://colorbrewer2.org

Figure 4.9: Color Schemes for Variable Representation

4.4.1 App Color Scheme

In Chapter 2.3 we mentioned that there is an environmental characteristic that highly impacts the usability of this system which is glare. Users of this system may want to consult this surf forecast visualization at the beach which is a place where we are exposed to bright sunlight. For that reason we decided to choose 5 different color schemes where there is a high contrast between the colors and also where colors are closely associated to the colors of the beach and ocean. In Figure 4.10 it is possible to see the 5 different color schemes that we chose.

Based on these different color schemes we proceeded to ask 15 different users two questions: which color scheme they liked the most and which color scheme they were able to visualize better under high sun exposure.

With the evidence of these results we chose the color scheme number where there is a high contrast between the colors and where users are most satisfied with. Figure 4.12 shows the 4 colors that were used in the components of the interface.

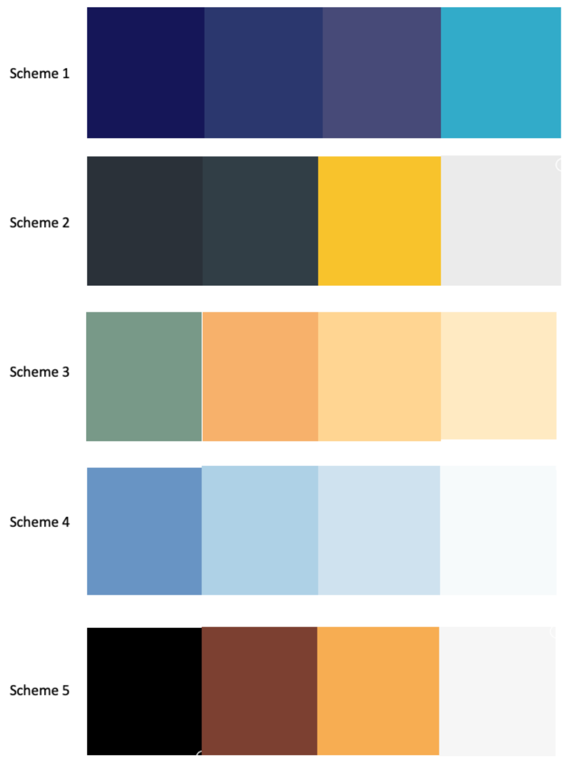
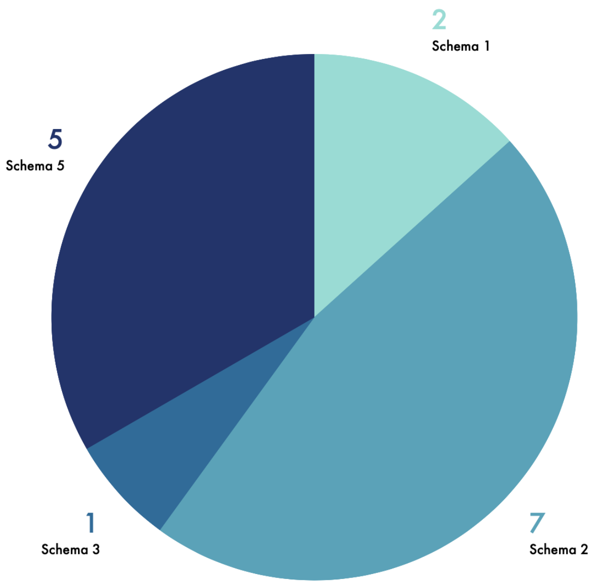
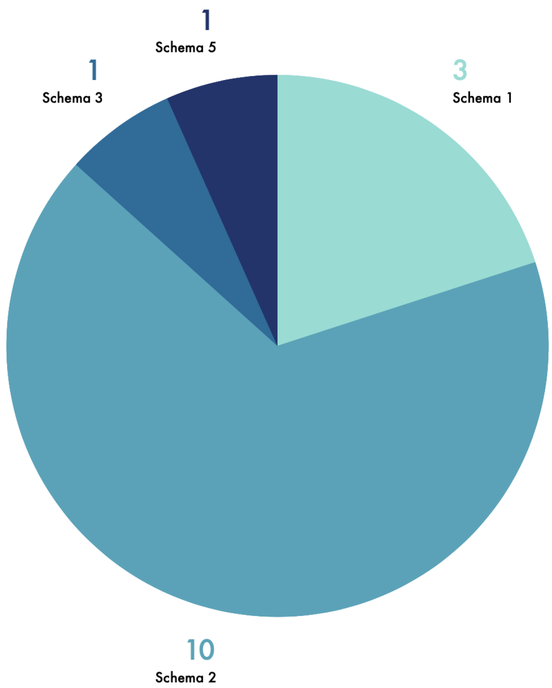


Figure 4.10: Color Schemes for User Testing



(a) Color Scheme Users Liked The Most



(b) Best Color Scheme Under Bright Conditions

Figure 4.11: User preferences of color schemes

Figure 4.12: App Color Scheme

5

Evaluation

Contents

Evaluating an interface is an essential step to make end products and services more usable and easier to learn for users. This chapter details the evaluation methods performed to evaluate the developed interface to guarantee that the system works as expected and satisfies the users' requirements. First we performed a functional evaluation of the system in order to understand if the application is responsive and therefore is adapting to different screen sizes and orientations. Next we did a system evaluation to understand how our application is performing regarding usability. We describe these two types of evaluation in the next subsections.

5.1 Functional Evaluation

Before analyzing how the system behaves in terms of functionality and usability we must evaluate if it works for the different browsers available and if its layout adjusts depending on the viewing screen size.

This evaluation was done by us, and focused on testing the behavior of the website for the most used browsers and screen resolutions.

Since it is not feasible to address all existent device's resolutions we must first analyze which are the most used screen resolutions for each platform (Desktop, Tablet and Mobile). In order to obtain this data we resorted to the website StatCounter[1] and the results were:

- **1** - **Desktop**. For the period between October 2019 and October 2020 the most used desktop screen resolution worldwide was 1366x768.
- **2** - **Tablet**. For the period between October 2019 and October 2020 the most used tablet screen resolution worldwide was 768x1024.
- **3** - **Mobile**. For the period between October 2019 and October 2020 the most used mobile screen resolution worldwide was 360x640.

Figures 5.1, 5.2, 5.3 show how the application behaves for each of the screen resolutions. Since on tablets and mobiles users can change the orientation of the device we have included both portrait and landscape screenshots of the application.

When comparing the interface on the different devices we are able to check how components adapt in order to better adjust to the screen's available space. The components on the top bar of the application decrease in size when moving from desktop to tablet and then to mobile view. It is even possible to see in the mobile portrait view that the input on the search component and the date on the calendar component disappear because there is not enough space (they can be seen whenever we click on each of the components). Also the zoom of the map component increases when we use smaller screen sizes.

[1] https://gs.statcounter.com

Based on these evidences we are able to say that this application is responsive since it adapts based on the screen size we are using.

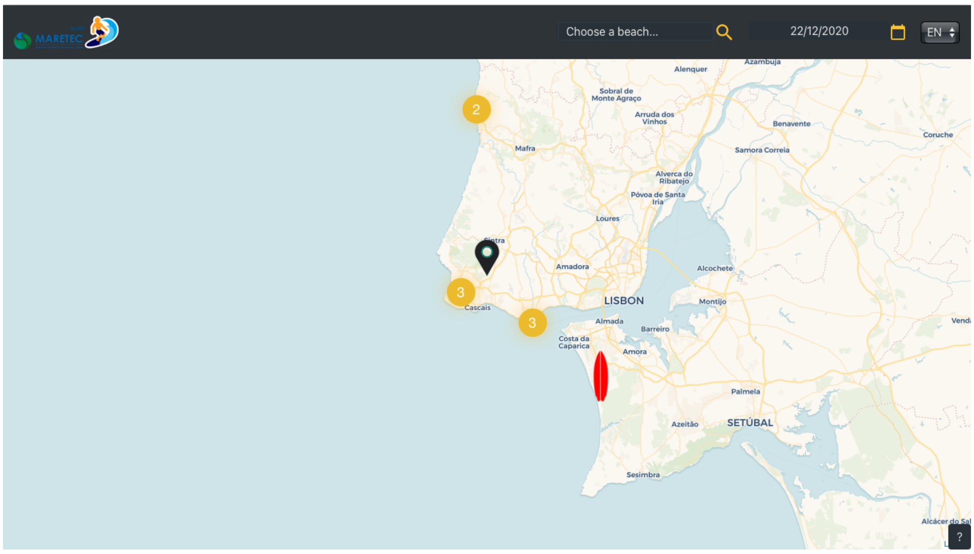


Figure 5.1: Desktop View of the Sea&Surf App

5.2 System Evaluation

System evaluation is an essential step since it helps us in identifying possible problems regarding the functionality and usability of our solution.

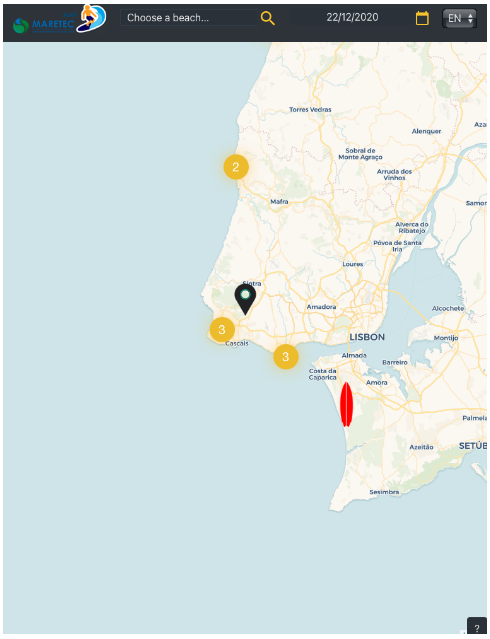
According to Fonseca, Campos, and Gonçalves [1], one of the methods that can be used to evaluate our system is User Testing.

5.2.1 User Testing

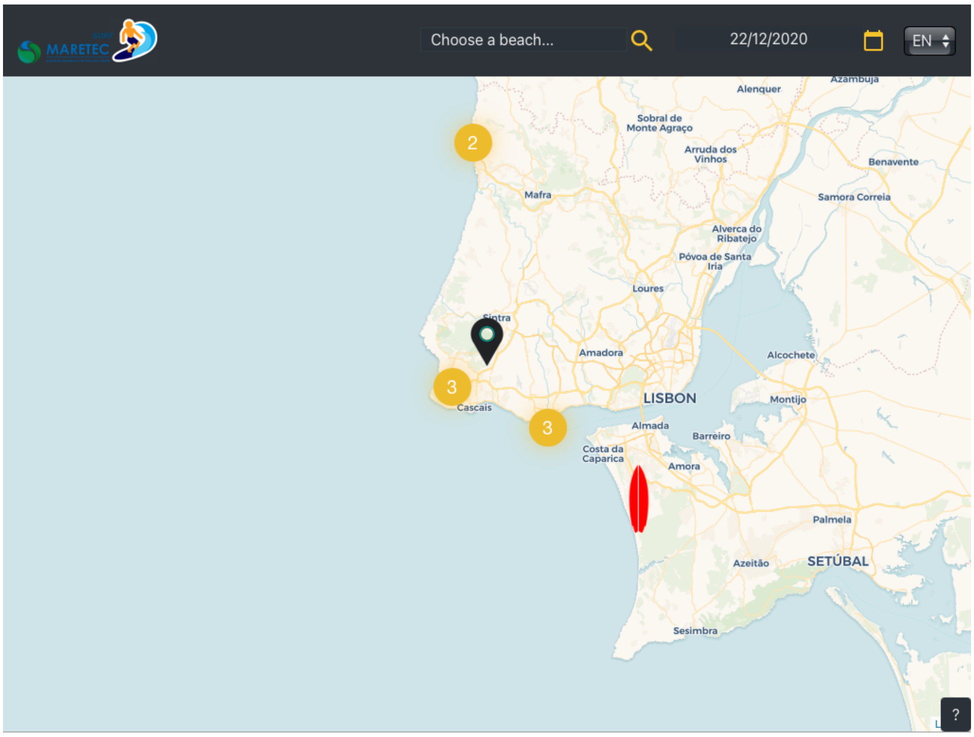
This is a technique that involves measuring the performance and satisfaction of users when performing certain tasks. The big advantage of this technique is that we deal with the end-users of our system, and they are the ones who can give us more information about the problems they may encounter.

For this technique, it is essential to develop a protocol to explain to the users that were going to test the application the conditions and the objectives of these tests. It is important for users to understand that we are testing the system and not them and therefore every difficulty or mistake they make, it is the system's fault.

In Appendix B it is possible to see the complete protocol which is divided in 5 different sections: Environment and Equipment; Introduction; Initial Information; Tasks; User Satisfaction Questionnaire.

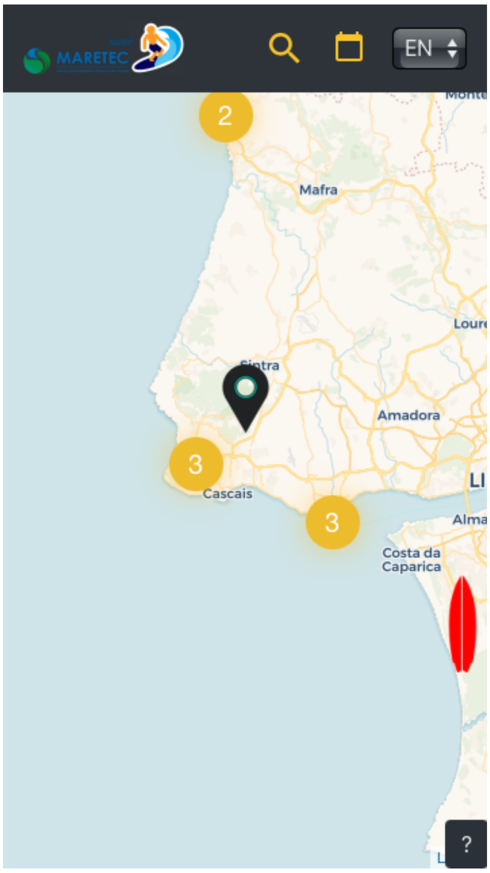


(a) Portrait View

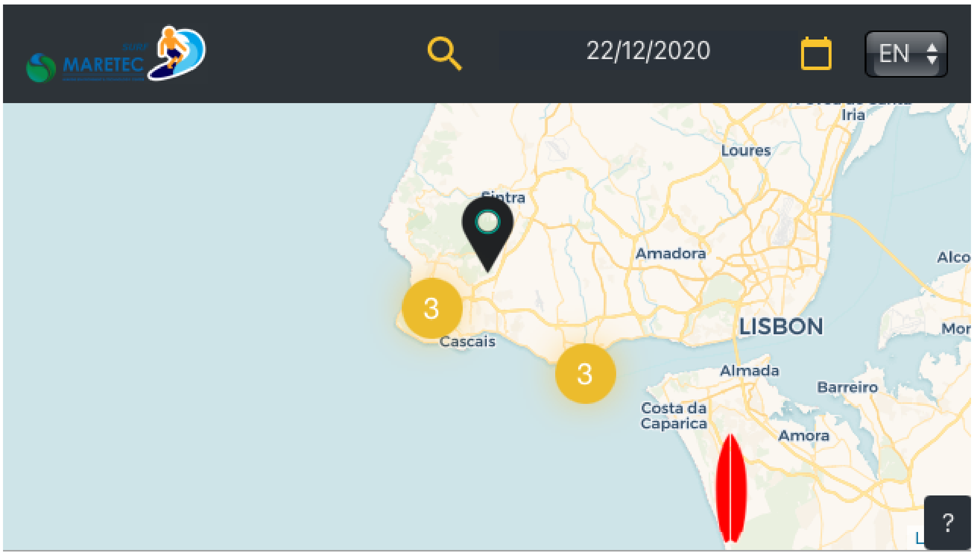


(b) Landscape View

Figure 5.2: Tablet View of the Sea&Surf App



(a) Portrait View



(b) Landscape View

Figure 5.3: Mobile View of the Sea&Surf App

- **1** - **Environment and Equipment**. Is the section where we state which are the conditions where we are performing the tests. Due to the current situation (Covid-19) it is impossible to guarantee the exact same conditions for all participants, but we will do our best to minimize the discrepancies between participants' conditions.
- **2** - **Introduction**. It is the section where we introduce the objective of the test to the participants. Users are informed they will need to perform 5 different tasks where each of them will be timed and checked for any errors. Users are informed that their identity will remain anonymous.
- **3** - **Initial Information**. This section is where we ask 7 questions in order to characterize the users, so that whenever we are analyzing results we can draw some conclusions based on users characteristics.
- **4** - **Tasks**. This section is where we define the 5 tasks that will be performed by users which are: Say which is the wave height for Praia de Carcavelos today at 3pm; Say which is the wave height for Praia de Carcavelos tomorrow at 3pm; Firstly select Portuguese as the defined language and then say which is the wave height for Praia do Guincho today at 4pm; Analyze the surf ranking for Praia de Carcavelos for today and say if it is good to go surfing; Without using the search bar choose Praia de Carcavelos and say which is the wave height today at 3pm.

  Each of these tasks involve success criterias (Efficiency, Effectiveness and Satisfaction) so that after we perform all the tests with the users we can conclude if the criterias were met and therefore if the application can be considered usable.
- **5** - **User Satisfaction Questionnaire**. In this section we used the System Usability Scale (SUS) developed by John Brooke. It is a questionnaire that allows developers to understand the overall usability of the system and the degree of satisfaction of users [1]. Based on the tasks performed using the developed solution users can answer ten questions on a scale of 1 to 5 regarding how much they agree with them [19]. Based on the answers, we can calculate a score that will let us know how our solution is behaving in terms of usability.

With this protocol we were able to carry out the tests with 20 different users with the following age distribution: 18 users between 19-34; 2 users between 35-54. In regards to the gender distribution 12 were male (60%) and 8 female (40%). All of them have practiced surf at least once in their lives, and the frequency which they surf nowadays is: 5 users rarely practice (25%); 7 users monthly (35%); 4 users weekly (20%); and 4 users everyday (20%). All users affirmed they use touch devices everyday and the distribution of operating system they use is: 11 users use IOS (55%); 9 users use Android (45%).

In Appendix C it is possible to see the data collected for each metric from each user in detail.

In Table 5.1 we show which are the criteria of success for each of the 5 tasks. Table 5.2 shows us the statistical data regarding the time in seconds to complete each task. With the information on both of these tables we are able to say if the success criteria of efficiency and effectiveness defined in the protocol were accomplished or not for each task.

Table 5.1: Criteria of success defined for each task.

	T1	T2	T3	T4	T5
Efficiency	Average under 20s	Average under 20s	Average under 30s	Average under 20s	Average under 15s
Effectiveness	90% under 25s	90% under 25s	90% under 35s	90% under 25s	90% under 20s
Satisfaction	Easy	Easy	Easy	Easy	Easy

The following list analyzes the values obtained in Table 5.2 in order to justify if the efficiency and effectiveness criteria are met for each task.

- **Task #1** - **Say which is the wave height for Praia de Carcavelos today at 3pm.** For this task we got an average of 16.1 seconds to complete the task which is under the 20 seconds mark defined for the efficiency criteria. We got 90% of the users under 23.1 seconds which is also under the mark of 25 seconds defined for the effectiveness criteria. Thus, we can say that for this task the criteria of efficiency and effectiveness were met.
- **Task #2** - **Say which is the wave height for Praia de Carcavelos tomorrow at 3pm.** For this task we got an average of 14.2 seconds to complete the task which is under the 20 seconds mark defined for the efficiency criteria. We got 90% of the users under 18.0 seconds which is also under the mark of 25 seconds defined for the effectiveness criteria. Thus, we can say that for this task the criteria of efficiency and effectiveness were met.
- **Task #3** - **Firstly select Portuguese as the defined language and then say which is the wave height for Praia do Guincho today at 4pm.** For this task we got an average of 18.2 seconds to complete the task which is under the 30 seconds mark defined for the efficiency criteria. We got 90% of the users under 25.4 seconds which is also under the mark of 35 seconds defined for the effectiveness criteria. Thus, we can say that for this task the criteria of efficiency and effectiveness were met.
- **Task #4** - **Analyze the surf ranking for Praia de Carcavelos for today and say if it is good to go surfing.** For this task we got an average of 19.1 seconds to complete the task which is under the 20 seconds mark defined for the efficiency criteria. We got 90% of the users under 30.1 seconds which is above the mark of 25 seconds defined for the effectiveness criteria. Thus, we can say that for this task the criteria of efficiency was met but the criteria of effectiveness was not met. This can be explained by the lack of knowledge of some users regarding the concept of surf ranking.
- **Task #5** - **Without using the search bar choose Praia de Carcavelos and say which is the wave height today at 3pm.** For this task we got an average of 10.4 seconds to complete the task

which is under the 15 seconds mark defined for the efficiency criteria. We got 90% of the users under 16.1 seconds which is also under the mark of 20 seconds defined for the effectiveness criteria. Thus, we can say that for this task the criteria of efficiency and effectiveness were met.

Table 5.2: Statistical Analysis of the time in seconds to complete each task.

	T1	T2	T3	T4	T5
Minimum	8	7	5	8	5
Maximum	26	55	34	35	18
Average	16.1	14.2	18.2	19.1	10.4
90th percentil	23.1	18.0	25.4	30.1	16.1
Standard Deviation	5.07	9.96	7.03	8.02	4.10

Table 5.3 shows us the statistical data regarding the errors on each task. This was an important step performed during the tests with the users because it helped us to identify several problems that our application has. Task #4 was the one were users made the most errors because since some of them were not familiarized with the concept of the surf's ranking, they ended up having more difficulty to understand it and finalize the task correctly. Even when users did not commit any mistakes, they would suggest us some fixes that would help them to better understand the flow of our application and at the end increase their user experience. The following list details the errors identified and the suggested fixing for each.

- **1** - The calendar only opens when users click on the icon. Several users committed the error of clicking on the date itself and the calendar did not open. This can be easily fixed by allowing users to click on both to open the calendar.

- **2** - The table with the information for the selected beach does not show the corresponding units for each variable, which does not help users when they are trying to interpret the data. This could be fixed by adding the units in front of each variable name.

- **3** - On smaller devices the table with the information gets shortened and a vertical scroll is added, but in some devices the scroll bar is hidden. Thus, users who have this problem do not know that there is more information on the table. This problem can be fixed by adding the scroll bar with a different color to all devices.

- **4** - Whenever users click on a surfboard icon to check the information for that beach instead of using the search bar, it is not clear which is the beach selected. An easy fix could be adding the name of the selected beach on top of the table with the information.

- **5** - It is not clear for some of the users which is the scale of the surf ranking. Although there is the help button that shows the scale and the corresponding colors, several users did not realize this was available and took more time to complete Task #4. This is something that is harder to be fixed, as it takes users to be more trained regarding the aspect of the surf rankings but we can help by adding a tooltip on the beach icon with the translation of the ranking to words (such as ”good” or ”average”).

Table 5.3: Statistical Analysis of the errors on each task.

	T1	T2	T3	T4	T5
Minimum	0	0	0	0	0
Maximum	0	1	2	2	1
Average	0	0.15	0.25	0.7	0.35
Standard Deviation	0	0.36	0.54	0.64	0.48

Table 5.4 shows us the statistical data regarding the degree of satisfaction of users on each task. With this information we are able to say if the last criteria of success (Satisfaction) is met for each task. As it can be seen on the table, all tasks scored above 4 in a scale of 0 to 5, which allows us to say that users were in general satisfied with the tasks and therefore the criteria of satisfaction was met for all tasks.

Table 5.4: Statistical Analysis of the degree of satisfaction on each task.

	T1	T2	T3	T4	T5
Minimum	5	4	4	3	4
Maximum	5	5	5	5	5
Average	5	4.95	4.7	4.35	4.8
Standard Deviation	0	0.22	0.46	0.65	0.4

Lastly, we used the Satisfaction Questionnaire to ask participants to score from 1 (strongly disagree) to 5 (strongly agree) the ten questions (available in Appendix B). Based on those scores we were able to calculate a SUS score by performing the following steps:

- **1** - For each odd numbered question, subtract 1 from the score.
- **2** - For each even numbered question, subtract their value from 5.
- **3** - Sum all the new values and multiply by 2.5.

Although it is not a percentage score it will give a score out of 100 which helps to understand better how our system is behaving in terms of usability. As it can be seen in Figure 5.5 this system obtained a score of 96.75 which according to Jeff Sauro [2] can be graded as an A+. What this means is that our system was evaluated by the users with a good usability and therefore we can say that our system was accepted by the users.

[2] https://measuringu.com/interpret-sus-score/

Table 5.5: Statistical analysis of the System Usability Scale scores.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
Score	5.0	1.15	4.9	1.05	4.7	1.2	4.8	1.05	4.75	1.0	96.75

Although our application failed to meet the criteria of effectiveness on Task #4, by successfully meeting all the other success criteria for each task and with a high SUS score we are in position to say that our system satisfied the user requirements defined in the start of this project, making this a successful project.

6

Conclusion

Contents

6.1 Final Remarks

With this project, we sought to take advantage of the already collected data from Maretec regarding both weather and maritime variables to provide surfers a solution for the visualization of forecasts oriented to the practice of surf. This work had the main goals of developing a responsive web application for displaying surf reports that would adapt to any viewing screen size and that would make available a ranking to help users to better understand the conditions of surf for each beach.

By dividing this project in 3 phases (Investigation, Development and Testing) we were able to reach a solid and optimized solution at the end. The Investigation phase turned out to be a major aspect of this project because it allowed us to construct a theoretical foundation around this work, where every decision made was clearly justified and compared with other alternatives. Also, by studying alternative solutions already existent on the market, we were able to understand what were the users requirements and what were their difficulties, which lead us to address both aspects on our solution. The testing phase lead us to quite exciting results, where we got a 96.75 score on the System Usability Scale, and only failed one criteria for one of the tasks. Although these are really good results we believe that there is still room for improvement, namely the fixing of the errors listed in Chapter 5 and other interesting features that were identified and are described in section 6.3.

Throughout the development of this project we faced several obstacles, mainly due to problems involving the server that runs the forecasts models, but with the help of the several people involved these obstacles were surpassed.

Overall, at the end of this work we believe that the main goals were achieved making this a successful solution that can be used by surfers.

6.2 System Limitations

This project has two main limitations that are outside of the scope of this project and therefore can not be controlled by us. The first one is the availability of the Maretec server to produce the forecasts everyday. This application only works if we have available the forecasts for every beach in Portugal, and therefore whenever the Maretec server is not producing the forecasts our application does not work. The second one is the availability of internet connection on user's devices. Since this is a web application, users need to have internet connection to be able to access the application.

6.3 Future Work

Throughout the development of this project several new ideas emerged that could highly increase the quality of this project. All the efforts were done to include some of those ideas, but since there is limited time for finishing this project there is still room for improvement in several areas of the application. The following list describes some of the ideas that could be done in a near future for this application:

- **Tuning of the surf ranking**. There is still room to increase the accuracy of this ranking by adding more parameters to the calculation, such as the tide. The current calculation uses a general algorithm for every beach, which in some cases might lead to wrong values since each beach is different and might behave differently for the same conditions.
- **Tides**. Since the tides are one of the conditions users look for, this is something that should be considered to be added in a future version of this project.
- **Live Webcams**. This is often a requested feature by surfers, since they like to actually see the live images of the beaches.
- **Favorites**. Users could save their favorite beaches in order to have a faster access to them, and consequently have a better user experience.

Bibliography

[1] M. J. Fonseca, P. Campos, and D. Gonçalves, "Introdução ao design de interfaces," *FCA-Editora de Informática*, 2012.

[2] M. Bano and D. Zowghi, "A systematic review on the relationship between user involvement and system success," *Inf. Softw. Technol.*, vol. 58, pp. 148–169, 2015.

[3] W. I. and A. Hammouri, "Responsive web design techniques," *International Journal of Computer Applications*, vol. 150, pp. 18–27, 09 2016.

[4] L. L. Giurgiu and I. Gligorea, "Responsive web design techniques," *International conference KNOWLEDGE-BASED ORGANIZATION*, vol. 23, 01 2017.

[5] B. S. Gardner, "Responsive web design: Enriching the user experience," *Sigma Journal: Inside the Digital Ecosystem*, vol. 11, no. 1, pp. 13–19, 2011.

[6] K. Meyer, "Toward a definition of minimalism: Principles of minimal visual design in web interfaces." 2015.

[7] F. Yang and F. Shen, "Effects of web interactivity: A meta-analysis," *Communication Research*, vol. 45, no. 5, pp. 635–658, 2018. [Online]. Available: https://doi.org/10.1177/0093650217700748

[8] K. Moreland, "Diverging color maps for scientific visualization," pp. 92–103, 2009.

[9] R. Borgo, J. Kehrer, D. H. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen, "Glyph-based visualization: Foundations, design guidelines, techniques and applications." pp. 39–63, 2013.

[10] M. Link, "A comparison of the frontend javascript gui frameworks angular, react, and vue," 2018.

[11] F. Almeida and J. Monteiro, "Approaches and principles for ux web experiences:: A case study approach," *International Journal of Information Technology and Web Engineering*, vol. 12, pp. 49–65, 04 2017.

[12] N. Jain, “Review of different responsive css front-end frameworks,” *Journal of Global Research in Computer Science*, vol. 5, no. 11, pp. 5–10, 2015.

[13] W. D. Cletus, A. Kakandar, and C. N. Paul, “Responsive web design frameworks: A review,” *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol*, vol. 4, no. 5, pp. 132–136, 2017.

[14] C. L. Mariano, “Benchmarking javascript frameworks,” 2017.

[15] E. Baralis, A. Dalla Valle, P. Garza, C. Rossi, and F. Scullino, “Sql versus nosql databases for geospatial applications,” in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 3388–3397.

[16] S. Venkatraman, K. Fahd, S. Kaspi, and R. Venkatraman, “Sql versus nosql movement with big data analytics,” *Int. J. Inform. Technol. Comput. Sci*, vol. 8, pp. 59–66, 2016.

[17] K. Nygård, “Single page architecture as basis for web applications,” G2 Pro gradu, diplomityö, 2015-08-24. [Online]. Available: http://urn.fi/URN:NBN:fi:aalto-201509184388

[18] S. C. Baker, “Making it work for everyone: Html5 and css level 3 for responsive, accessible design on your library’s web site,” *Journal of Library & Information Services in Distance Learning*, vol. 8, no. 3-4, pp. 118–136, 2014.

[19] J. Brooke *et al.*, “Sus-a quick and dirty usability scale,” *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.

A

Interviews with Surfers

Surfer 1.

1. What websites/apps do you use to check the waves forecast?

   • ”The websites I use to check the waves are: Beachcam and Windguru. If Windguru is not available I use Magicseaweed.”

2. What is your first impression when you open the website(s)/app(s)?

   • ”Regarding Beachcam: it is really annoying because it makes us watch ads for a long time, and sometimes I’m in a hurry and have to wait for it to finish the ad. Regarding Windguru: it is a little bit messy because it has a lot of information in vertical and horizontal lines. For those less experienced Windguru is worthless.”

3. Why do you use that website(s)/app(s)?

   • ”First because the ocean is unpredictable. Us (surfers) can never actually know what we can count on. It depends on a lot of things like the tides, the moon cycles, the wind, the swell, etc.

These websites allow us to understand if we can surf or teach surf lessons. It helps organize our day and work."

4. What devices do you use to access it?

   - "I mainly use my mobile phone and my computer."

5. When and where do you access it?

   - "Normally at home or school, and before I plan to surf or schedule a class. I usually check one day before and a couple of hours before since it is when the conditions are the most accurate."

6. Do you consider it user-friendly?

   - "I think Beachcam is user-friendly because it allows to choose the beach we want from North to South and is easy to navigate. The only downside is the ads. Windguru as I already said it is complicated to understand all the information that is being immediately shown, so it is not user-friendly."

7. Does it have everything you need? If not, what's missing?

   - "Beachcam does not give us the forecast for next hours/days. It only allows for those who want to check the current behaviour of the waves. Windguru has all the information I need but lacks the visual part where I can actually see the waves and make my personal judgement."

8. Would you change anything on it?

   - "Beachcam has a nice design. The black color makes it cleaner and the information stands out more effectively. But I would add more information to it. Windguru is less cleaner and the design is not attractive. Windguru also gives too much information especially regarding the range of days."

9. Would it help if you had access to the history of the previous days?

   - "Yes, although it is not needed to know if there are good conditions to go surfing, sometimes when I'm talking to my students I show them conditions of previous days where they had class with me, to explain to them how to correctly analyze reports."

10. What are the positive aspects? And negatives?

    - "Beachcam: the positive is that shows live conditions of several beaches in Portugal. The negative is that is does not show forecasts for the next few days. Windguru: positive is that it forecasts for the next few days, and the negative is that the information is shown in a tabular way with a lot of lines and columns."

11. Was it easy to understand when you saw it for the first time?

    - "Beachcam yes, it is pretty straightforward for new users. Windguru is pretty difficult for those less experienced since they need to join all the factors to understand if there are good conditions."

Surfer 2.

1. What websites/apps do you use to check the waves forecast?

    - "Windguru to check the size of the waves, wind and directions; and Instituto Hidrografico to check the tides."

2. What is your first impression when you open the website(s)/app(s)?

    - "Windguru: really hard to understand and to join all the information. IH: easy but annoying since it needs refreshing every time you go there."

3. Why do you use that website(s)/app(s)?

    - "I use both of them because I need to know all the information to know the conditions for surfing."

4. What devices do you use to access it?

    - "I use my mobile phone 99.99% of the time. Only use PC if I'm already using it for other stuff."

5. When and where do you access it?

    - "Everywhere, but mostly at home. Normally I check it during the weekend to know the conditions for the week and then daily to be more accurate."

6. Do you consider it user-friendly?

    - "Windguru: not user-friendly at all, there is too much information and it is not well organized. IH: Easy to navigate but has a boring design."

7. Does it have everything you need? If not, what's missing?

    - "Windguru: has everything I need regarding waves and winds, but it could have tides also. It does not have because is a website more suited for kitesurf and windsurf. IH: has everything I need."

8. Would you change anything on it?

    - ”Windguru: I would change the whole layout to be more understandable. I would add more visual elements like colors in the different winds or waves. I would also add a feature to tell which beach is going to have better waves in a specific region according to the swell and wind. IH: I would change the design to something more appealing.”

9. Would it help if you had access to the history of the previous days?

    - ”Yes, sometimes it is helpful to compare to previous days to know if the swell is going to stay the same or change.”

10. What are the positive aspects? And negatives?

    - ”Windguru: positive aspect is the completeness of information that it gives. The negative is the difficulty of understanding all the data. IH: fast and easy to navigate but not appealing.”

11. Was it easy to understand when you saw it for the first time?

    - ”Windguru: really really hard for beginners. IH: easy, everything is intuitive.”

Surfer 3.

1. What websites/apps do you use to check the waves forecast?

    - ”I use Windguru, beachcam e windy.”

2. What is your first impression when you open the website(s)/app(s)?

    - ”From the three the one that gave me the best impression was Windy since it has a cleaner and nicer design.”

3. Why do you use that website(s)/app(s)?

    - ”I think they are the ones who give me a better perception of the sea conditions and which beach is the best to surf for the day I’m looking at. Also, these are the ones I am more used to and are easier for me to use.”

4. What devices do you use to access it?

    - ”Smartphone and rarely laptop.”

5. When and where do you access it?

    - ”At home.”

6. Do you consider it user-friendly?

   - ”Beachcam and Windy are pretty intuitive. Winguru requires knowledge that beginners don’t have, but afterwards it becomes easier.”

7. Does it have everything you need? If not, what’s missing?

   - ”The conjugation of the 3 gives me everything. Of course, for example I trust Windy more than Windguru to check on the wind for beaches that I don’t know. On the other hand I trust Windguru more for waves and tides.”

8. Would you change anything on it?

   - ”Regarding beachcam I would change the representation of the tides to indicate what are the variations compared to the average sea level.”

9. Would it help if you had access to the history of the previous days?

   - ”Yes a lot, mainly when I go to new beaches where I don’t know the optimal conditions.”

10. What are the positive aspects? And negatives?

    - ”They allow the study of the best beach to go surfing without having to actually go to the beach.”

11. Was it easy to understand when you saw it for the first time?

    - ”Beachcam yes. Windguru no. Windy more or less but I already had experience in understanding the conditions.”

Surfer 4.

1. What websites/apps do you use to check the waves forecast?

   - ”I use Windguru, windy (app), and meo beachcam.”

2. What is your first impression when you open the website(s)/app(s)?

   - ”The information is well organized for those who are already familiarized with surf reports. However there is a lot of information that is irrelevant for the practice of surf.”

3. Why do you use that website(s)/app(s)?

   - ”Windguru- because it seems the most accurate; Windy- because gives time and height of the tides; Beachcam- because of the image streaming.”

4. What devices do you use to access it?
    - ”Always my smartphone.”
5. When and where do you access it?
    - ”At home or outside before going to the beach.”
6. Do you consider it user-friendly?
    - ”Yes, all of them are intuitive for me (experienced user).”
7. Does it have everything you need? If not, what’s missing?
    - ”Windguru is missing information related to the tides. And it would also benefit with the use of a map in order to represent the wind and waves direction. Windy should be more accurate.”
8. Would you change anything on it?
    - ”I would not change anything except the missing information I referred previously.”
9. Would it help if you had access to the history of the previous days?
    - ”Yes there are some cases where history would help in better understanding the forecasts.”
10. What are the positive aspects? And negatives?
    - ”Positive: easy access, fast, free and transmit the information in an organized way. Negative: Sometimes forecasts are not accurate. Webcams sometimes do not work.”
11. Was it easy to understand when you saw it for the first time?
    - ”Windguru and Windy- took me a little to get used to the technical details. Beachcam- pretty straightforward.”

B

User Testing Protocol

B.1 Environment and Equipment

Since we are currently experiencing a pandemic (Covid-19) and all contact between people must be avoided, the following test will have to be carried out differently than usual. For this, and whenever possible this test will be carried out under the supervision of the organizer through a video call using any of the existing platforms (Zoom, Teams, etc.) where the participant's screen can be shared with the organizer. As it is not possible to guarantee exactly the same conditions for all participants, they will have to use their personal mobile phone, as long as they have access to the internet. In addition, a stopwatch will be used by the organizer to measure the time that the user takes to complete each task.

B.2 Introduction

The introduction of the tests to the users is as follows:

First of all, we would like to thank you for participating in this test, as it will help us a lot in the development of this project.

This test will consist on the execution of 5 tasks mentioned below in order to measure whether the application meets the expectations and the objectives initially outlined. If you have any questions, you should place them to the organizer before or after each task, since during the tasks you will not be able to get any help. When carrying out each task, you may take as long as you think necessary to complete it successfully. If you feel that you are unable to complete the task, you must indicate it to the organizer and this task will be stopped.

It is important to note that we are testing the application and not the user, and that you can abandon the test at any time if you wish to do so.

All information collected here will support the development of the project and the identity of the participant will remain anonymous.

B.3 Initial Information

The following questions are useful to better understand some of the characteristics of our users and to draw some conclusions based on their performance.

Question #1: What is your age group?

Possible Answers: Under 18; 19-34; 35-54; 55-74; Above 74.

Question #2: What is your gender?

Possible Answers: Male; Female.

Question #3: Have you ever practiced surf?

Possible Answers: Yes; No.

(Skip to question #5 if you answered No)

Question #4a: How often do you surf?

Possible Answers: Everyday; Weekly; Monthly; Rarely; Never.

Question #4b: Do you normally access websites to check the sea conditions?

Possible Answers: Yes; No.

Question #5: How often do you use touch devices?

Possible Answers: Everyday; Weekly; Monthly; Rarely; Never.

Question #6: Which operating system do you use?

Possible Answers: IOS; Android; Other.

B.4 Tasks

Task #1: Say which is the wave height for Praia de Carcavelos today at 3pm.

Criteria:

Efficiency - In average uses perform this task in less than 20 seconds.

Effectiveness - 90% of users finish this task in less than 25 seconds.

Satisfaction - Users considered this task easy.

Task #2: Say which is the wave height for Praia de Carcavelos tomorrow at 3pm.

Criteria:

Efficiency - In average uses perform this task in less than 20 seconds.

Effectiveness - 90% of users finish this task in less than 25 seconds.

Satisfaction - Users considered this task easy.

Task #3: Firstly select Portuguese as the defined language and then say which is the wave height for Praia do Guincho today at 4pm.

Criteria:

Efficiency - In average uses perform this task in less than 30 seconds.

Effectiveness - 90% of users finish this task in less than 35 seconds.

Satisfaction - Users considered this task easy.

Task #4: Analyze the surf ranking for Praia de Carcavelos for today and say if it is good to go surfing.

Criteria:

Efficiency - In average uses perform this task in less than 20 seconds.

Effectiveness - 90% of users finish this task in less than 25 seconds.

Satisfaction - Users considered this task easy.

Task #5: Without using the search bar choose Praia de Carcavelos and say which is the wave height today at 3pm.

Criteria:

Efficiency - In average uses perform this task in less than 15 seconds.

Effectiveness - 90% of users finish this task in less than 20 seconds.

Satisfaction - Users considered this task easy.

B.5 User Satisfaction Questionnaire

System Usability Scale Questionnaire	Strongly Disagree				Strongly Agree
1. I think that I would like to use this product frequently.	1	2	3	4	5
2. I found the product unnecessarily complex.	1	2	3	4	5
3. I thought the product was easy to use.	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this product.	1	2	3	4	5
5. I found the various functions in the product were well integrated.	1	2	3	4	5
6. I thought there was too much inconsistency in this product.	1	2	3	4	5
7. I imagine that most people would learn to use this product very quickly.	1	2	3	4	5
8. I found the product very awkward to use.	1	2	3	4	5
9. I felt very confident using the product.	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this product.	1	2	3	4	5

Figure B.1: System Usability Scale Questionnaire

C

User Testing Results

Table C.1: Time in seconds for each user to complete each task.

	Task #1	Task #2	Task #3	Task #4	Task #5
User #1	12	14	18	13	9
User #2	14	11	25	11	12
User #3	24	55	34	10	5
User #4	10	10	29	35	16
User #5	21	18	24	30	18
User #6	11	12	9	18	6
User #7	21	10	24	30	9
User #8	13	9	23	19	17
User #9	23	11	14	13	16
User #10	26	10	17	31	12
User #11	16	13	18	19	14
User #12	17	16	19	23	12
User #13	8	4	5	10	6
User #14	19	15	20	25	6
User #15	18	14	15	10	5
User #16	12	15	12	16	10
User #17	11	10	7	18	7
User #18	20	12	21	27	10
User #19	15	18	14	15	6
User #20	11	7	16	8	11
Average	16.1	14.2	18.2	19.1	10.4
Standard Deviation	5.07	9.96	7.03	8.02	4.10

Table C.2: Number of errors for each user on each task.

	Task #1	Task #2	Task #3	Task #4	Task #5
User #1	0	0	0	0	0
User #2	0	0	1	1	0
User #3	0	0	0	1	1
User #4	0	0	1	2	0
User #5	0	1	2	1	1
User #6	0	0	0	0	0
User #7	0	0	1	1	1
User #8	0	0	0	1	0
User #9	0	0	0	1	0
User #10	0	0	0	2	1
User #11	0	0	0	0	1
User #12	0	1	0	0	0
User #13	0	0	0	0	0
User #14	0	0	0	1	1
User #15	0	0	0	0	0
User #16	0	0	0	0	0
User #17	0	0	0	1	1
User #18	0	0	0	1	0
User #19	0	1	0	1	0
User #20	0	0	0	0	0
Average	0	0.15	0.25	0.7	0.35
Standard Deviation	0	0.36	0.54	0.64	0.48

Table C.3: Degree of satisfaction for each user on each task.

	Task #1	Task #2	Task #3	Task #4	Task #5
User #1	5	5	5	5	5
User #2	5	5	4	4	5
User #3	5	4	4	4	5
User #4	5	5	5	3	4
User #5	5	5	4	3	4
User #6	5	5	5	4	5
User #7	5	5	4	4	5
User #8	5	5	4	5	4
User #9	5	5	5	5	5
User #10	5	5	5	4	5
User #11	5	5	5	5	5
User #12	5	5	4	5	5
User #13	5	5	5	5	5
User #14	5	5	5	4	5
User #15	5	5	5	5	5
User #16	5	5	5	5	5
User #17	5	5	5	4	5
User #18	5	5	5	4	5
User #19	5	5	5	4	5
User #20	5	5	5	5	4
Average	5	4.95	4.7	4.35	4.8
Standard Deviation	0	0.22	0.46	0.65	0.4

Table C.4: System Usability Scale scores for each user.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
User #1	5	1	5	1	5	1	4	1	5	1	97.5
User #2	5	2	5	1	5	1	4	1	5	1	95
User #3	5	2	5	1	5	1	5	1	5	1	97.5
User #4	5	1	5	1	5	1	5	1	5	1	100
User #5	5	1	5	1	5	1	4	1	4	1	95
User #6	5	1	5	1	4	2	5	1	5	1	95
User #7	5	2	4	1	5	1	5	1	4	1	92.5
User #8	5	1	5	1	5	1	5	1	5	1	100
User #9	5	1	5	1	5	1	5	1	5	1	100
User #10	5	1	4	2	4	1	5	2	4	1	87.5
User #11	5	1	5	1	5	2	5	1	5	1	97.5
User #12	5	1	5	1	5	2	5	1	5	1	97.5
User #13	5	1	5	1	5	1	5	1	5	1	100
User #14	5	1	5	1	5	1	4	1	5	1	97.5
User #15	5	1	5	1	5	2	5	1	5	1	97.5
User #16	5	1	5	1	4	1	5	1	5	1	97.5
User #17	5	1	5	1	5	1	5	1	5	1	100
User #18	5	1	5	1	4	1	5	1	4	1	95
User #19	5	1	5	1	4	1	5	1	5	1	97.5
User #20	5	1	5	1	4	1	5	1	4	1	95
Average	5.0	1.15	4.9	1.05	4.7	1.2	4.8	1.05	4.75	1.0	96.75
Standard Deviation	0	0.36	0.30	0.22	0.46	0.40	0.40	0.22	0.43	0	

D

Surfboards

There are 3 characteristics of the surfboard[1] that influence the surfers' performance: its length, width, and depth.

Length. Longer boards make it easier to paddle and to stand up on since they are more stable. These boards are normally used by beginners since they have more difficulties in getting up when riding a wave. On the other hand, these boards are less maneuverable which makes it more difficult for surfers to perform tricks with it.

Width. Wider boards have higher buoyancy in the water. As well as the length, the wider the board the more stable it is.

Depth. Thicker boards float better, making the ride smoother and easier to paddle. Usually, the heavier the surfer is the thicker the board should be.

Thus these 3 characteristics combined create different kinds of surfboards. The most common are the shortboards, fish, gun, and longboards. All these types of surfboards[2] are made of polyurethane or polystyrene foam covered with layers of fiberglass cloth, and polyester or epoxy resin. These materials

[1] https://en.wikipedia.org/wiki/Surfboard
[2] https://clubofthewaves.com/feature/types-of-surfboard/

are what make the boards light and resistant.

It is also really important to have a notion about the different parts[3] of a surfboard since over the rest of the document some of those terms will be mentioned.

Bottom. It is the underside of the board and what is in touch with the water. It is mostly concave in order to facilitate the flow of water when riding waves but it can also be convex (normally used on more traditional boards) which move more water and sit lower in the wave.

Deck. It is the upper side of the board and where the surfer stands on. As the bottom part of the board, the deck is normally concave. Surfers apply wax to this area of the surfboard in order to help them stick to it while surfing and don't slip.

Nose. It is the front part of the board. It can either be pointy or rounded and is slightly curved (See "Nose Rocker" in Figure D.1) to help to perform maneuvers.

Tail. It is the end part of the board. It can have many shapes (square, rounded, swallow, etc.) that influence how a board reacts on the surface of the wave. More angular shapes help to perform sharper turns whereas rounder shapes are useful for more smoother turns.

Fins. It is a stabilizing rudder located under the board at the back. This is what gives stability to the surfboard and what prevents it from sliding when performing turns and tricks on the wave. Nowadays it is possible to have two (twin-fins), three (tri) or four fins (quad). Normally these are detachable in order to allow the variation of surf styling.

Leg Rope. Also called leash, is the plastic cord that links the surfer to the board. It is mainly useful to prevent the board from being taken away from the surfer by waves and hitting other surfers.

Rails. It is the edges of the board. They are useful since surfers can grab them to perform turns on the wave face. Rounded rails are called "soft" and adhere better, while squared rails are called "hard" and are more maneuverable.

Rocker. It is the bottom curve of the board. It directly influences factors such as speed or maneuverability. The curvier it is the more maneuverable it will be but will ride less fast. The flatter it is it will be faster but less maneuverable. Usually, a 50/50 rocker is the go-to option.

[3] https://calimasurf.com/news/guide-learn-surf

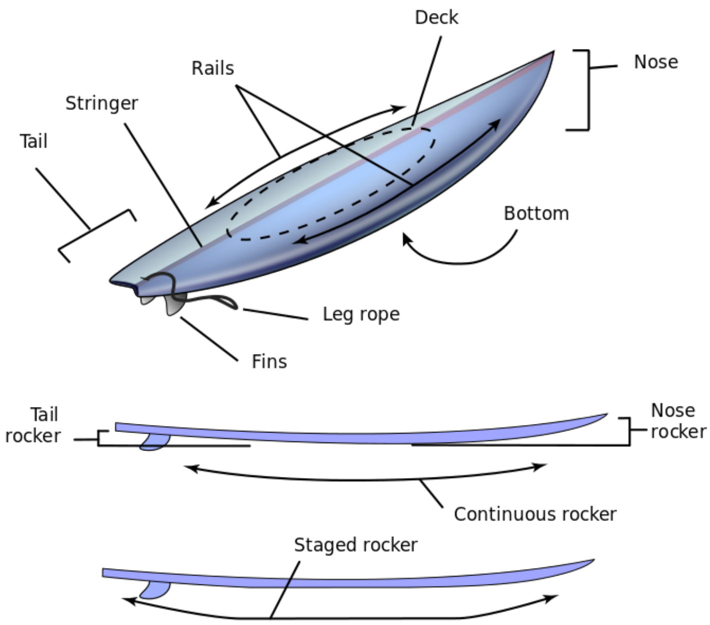


Figure D.1: Different components of a Surfboard, source: surfnation.com