

Dashboard for Ubiquitous Analytics

Mariana Santos

Instituto Superior Técnico

Lisbon, Portugal

marianacarapitosantos@tecnico.ulisboa.pt

ABSTRACT

As Mark Weiser predicted in 1991, computer access is everywhere, completely embedded in our everyday lives. They are always in the background, not requiring active attention, but data is always ready to be used. All these devices, collecting and presenting data from our environments, make that not only computing is ubiquitous but also the data. Embedding this data in our environment amplify human cognition, since the data is part of our physical world and consequently part of our cognition process. Public displays are one of the most ubiquitous elements of our visual culture, being completely ingrained in their specific physical, social, and cultural setting. However, they should not disappear in the background. Instead, they should become the backbone of a new global communication medium. The goal is to develop a dashboard to be deployed on displays around the Madeira Islands that give passers-by, locals or tourists, relevant information about the place. A dashboard that becomes part of the environment, but that stands out enough to be useful. This work was developed following an iterative and incremental methodology where user tests were used to validate the system.

KEYWORDS

Ubiquitous Computing; Ubiquitous Analytics; Public Display; Mobility; Human-Data Interaction

1 INTRODUCTION

In 1991, Mark Weiser, who is considered the father of ubiquitous computing, stated that the most important technologies are the ones that disappear; the technologies that are so embedded in our everyday life that they become indistinguishable from it. Their constant background presence does not require active attention, but the information is ready to use at a glance [14].

From 2013 to 2019, the web traffic that comes from mobile phones grew from 16% of the worldwide traffic to 52%. The idea of ubiquitous computing that Mark Weiser had is now a reality. The majority of people, even more in developed countries, have a mobile phone with Internet access and, as previous numbers state, increasingly use them to access the available information online. Computing access is everywhere: in walls, on our wrists, on our pockets. All these devices are right there, about to be used as needed and that

demands that the data resources need to be available anytime and anywhere.

All these networked devices, from smartphones and tablets to large displays and tabletops, can be combined into a single environment. The ensemble of these multiple heterogeneous sources, which can also be distributed in time and in space, provides a more complete data set. Allowing the accessibility to these deep and interactive analytics by several individuals with different backgrounds leads us to ubiquitous analytics [3]. Ubiquitous analytics amplify human cognition by embedding the analytics into smartphones, tablets, laptops, tabletops, and large displays. That will make them part of the physical world and consequently part of our cognition process.

Public displays, of all kinds, are one of the most ubiquitous elements of our visual culture. From road signs and billboards on the streets to posters on stores or notice boards in schools, as we look around in our environment we are constantly facing a very broad range of public displays with multiple forms and serving very different purposes. Most of these displays are deeply embedded in their specific physical, social and cultural setting and they are a crucial part of the already established practices [5]. Although this situatedness is necessary for public displays to fulfill their purpose, it also leads to the disadvantage of making them disappear in their environment and consequently being ignored.

However, public displays should not disappear in the background. Instead, they should become the backbone of a new global communication medium, as in the ubiquitous computing vision Weiser had [2].

Our project builds on previous work at Interactive Technologies Institute in the context of the Beanstalk project. Beanstalk is a multidisciplinary project with the aim of designing and delivering a product to the board of the Tourism of Madeira that allows them to analyze trends in tourism and marketing, that can stimulate the local economy.

Goals

The main goal of this work is to create a dashboard with ubiquitous data that the viewers can interact with, in a way that the dashboard and its analytics are part of their environment. That dashboard will be deployed on public displays scattered around the Madeira Islands. The objectives of the dashboard are:

- The dashboard should give the power to change its content to viewers, based on the information they find relevant in the moment;
- The communication between the viewer and the dashboard should be done through the viewer's mobile phone;
- The development of the system and the personalization of the dashboard has to be developed in a way that does not disclose revealing content or personal identity;
- If the dashboard receives contradictory viewers' preferences, it should have a system to choose which is the best option to show at a certain time;
- The dashboard should reflect the environment and the people in the environment where the display will be deployed in, in our case, the Madeira Islands.

2 RELATED WORK

In order to design content and applications for public displays that are more likely to serve their intended purpose, belong to the environment but also stand out enough to be relevant, an audience and context should always be set. Then, already having these premises in mind, there are two major things that should be taken into consideration and that are relevant for this work: content and situatedness.

When it comes to the type of the content, what we can find on public displays is varied, from art and education to news and advertising, advertising being the most common [4].

Regarding content format, we can find many different examples: text, animated text, still images, animations, sound, holograms, 3D images, projections. Although video format is the one viewers find more attractive [4], viewers are more likely to spend time looking at physical signage. This is explained by the fact that, while digital signage is more attractive and dynamic, users prefer to have control in what they want to see at the moment. This is an important fact to have present and the main argument that supports the need for personalization.

There are also some other discoveries from extensive investigations that are important to consider when creating content for this type of medium. Viewers decide extremely fast if the content of a display has relevance or not. This results in content that requires more than a few seconds to absorb and understand is likely to be ignored [4].

When it comes to situatedness, the majority of large displays in public areas are designed without proper consideration for the space where they are deployed in, the people around and the local activities [4]. In turn, that may affect the relevance of those displays, leading to lower utility and less attention. Digital displays should be designed to reflect both the environment in which they are placed and the actions of the people around them.

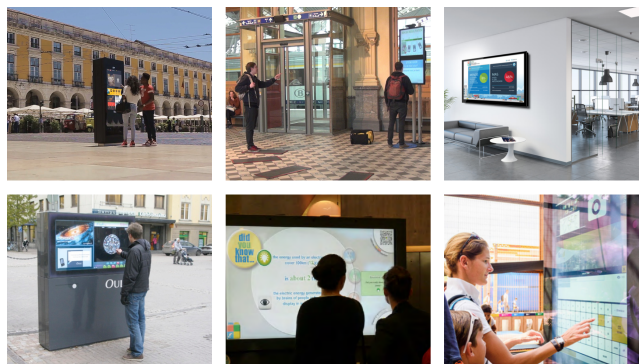


Figure 1: Examples of public displays

There are six dimensions that represent the main sources of situatedness for a public display: location, spatial association, activity, community, perceived ownership and place identity [5]. Location, activity, community and place identity are the ones related to the development of the content itself and that are more relevant to this work.

It is known that situatedness in digital displays can not be grounded in the exact same assumptions as nondigital ones and we can't simply do a direct translation between them. Therefore, the goal is to find new processes that were not previously supported by nondigital displays but are supported by digital ones, as these are more dynamic, more prone to being interactive and more adaptive to different situations around them.

Despite the amount of research done in the field, a structured framework on how to approach situatedness for digital displays is still needed. In the meantime, while developing content for them, we need to take into consideration these dimensions in order to make sure that the displays are useful in the environment and culture they are deployed in.

Content scheduling

Content scheduling is the art of knowing when and where to show a particular content item. It is a challenging question: it is necessary to understand the context of displays and viewers and it is not an easy task to select content items and choose the right schedule for them in the individual display and across the entire display network [9].

Typical scheduling systems are limited regarding changing the content schedule based on external factors at the same time that large sets of requirements and constraints need to be taken into account. Thus, the creation of scalable, automated scheduling systems, that determine which content should be shown based on the audience, location, and other contextual information, is one of the most important challenges for future display environments [9].

Human-display interaction

When creating content and design for public displays, guaranteeing utility, usability and likability is not enough. Displays need to grab the attention of passers-by, motivate citizens to interact with them and tackle the issues of interaction in public [10]. The reason for this, is that interaction with public displays does not start with the interaction itself. The viewer is just passing by and its interaction needs to be attracted.

Interaction can be divided into two types: implicit interaction and explicit interaction. The implicit one provides a “zero-click” model, where displays react to users passing by. In explicit interaction, users decide to interact with the content that is being currently displayed. It includes several techniques, like mediated (through a mobile phone, for example) and direct interaction (like a touch screen) [2].

Michelis at al. [7] modeled interaction focusing on audience behavior and called it “The Audience Funnel”. The model starts with people just passing by and is followed by them looking at the display and/or reacting to it. Then, a subtle interaction happens, only when is possible to interact. After this, a more direct interaction occurs. Finally, people can take follow up actions, like taking pictures, for example.

The major challenge is to overcome the first step: catch the attention of passers-by. In order to overcome the following ones, people must be motivated.

3 DASHBOARD SYSTEM

Data

The data that we will be using was collected as part of research to understand the flow and dynamics of tourism in Madeira Islands, in the context of the Beanstalk project. The data was gathered through a Wi-Fi monitoring infrastructure, with routers placed in different locations in Madeira in order to be able to capture how many people passed by that specific location [8].

To populate the dashboards we consume data from the endpoints made available to access that data. To make this possible, every screen is linked to its nearest router. The information captured by that router is the one that will be displayed on that screen. Besides the router, every screen has also a location name and POI identifier, which are also used as reference points. POI stands for point of interest.

Low and High Fidelity Prototyping

We started the development of this work by prototyping. Prototyping provides a communication basis for discussions among all the groups involved in the development process. Prototypes help in identifying difficulties, clarifying problems, and arriving at design decisions, before too many resources are used [6].

The first step to handle was the low-fidelity prototype. In figure 2 is the low-fidelity prototype developed for this work that allowed us to check if the design fulfilled the requirements and in what ways we could visualize the data we had. The designs were developed taking into account the data that we had, that was gathered previously.

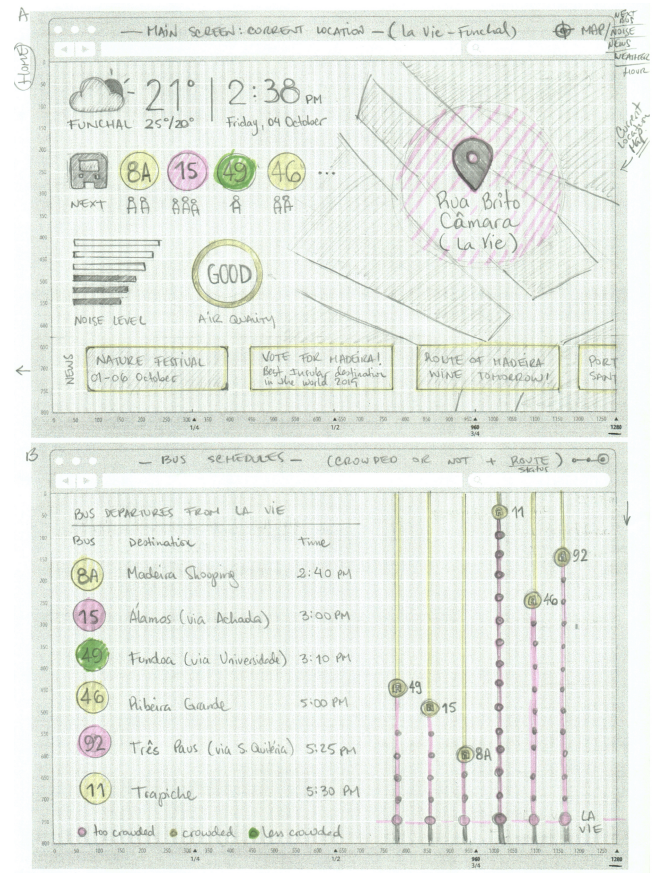


Figure 2: Low-fidelity prototype developed for the project

As we can see on figure 2, the main screen contains a map, which gives the display a location dimension as source of situatedness. Also on main page, we have information regarding buses and events, that represent an activity dimension, contributing to situatedness also.

The flow between the several pages will be defined by the interaction with the viewers. That interaction will be explicit, the viewers are just passing by and they are the ones that decide to interact with the display, and will be mediated through their mobile phones. This interaction will work as content scheduler: the content is changed based on the preferences that the views communicate to the public display.

Since the visual component of this project is crucial for its success, after validating the LFP, the team developed the high-fidelity prototype (HFP), which is very similar the functional prototype.

Functional Prototype

The last step was to develop a functional prototype, which is a prototype much similar to the final product, that captures both its appearance and functionality.

Setup. As previously mentioned, the goal of the dashboard is to be deployed on different public displays scattered around Madeira Islands. As such, the information that each one of the displays will be presenting will be related to the place where the display is situated. To achieve this, we need to complete a list of steps so the dashboard can get the right information for the place it is at.

Firstly, we need to create a CSV with the displays' latitude and longitude. Each line represents the exact location of a different display. When the file is ready, we can run a task that will create these displays on the database, give each one of them an identifier and associate them with the closest router, the name of the location and the POI identifier of the place they are deployed at.

Then, another task should be run in order to import the data of the buses to the database and associate each bus stop to the nearest displays. The data for the buses is consumed from the endpoints mentioned above. It is being imported to the database and not used directly from the endpoint, unlike the rest of the data, because there is too much of it and would take a long time to process in the runtime. Storing it in the database, the relations between the displays and the corresponding bus stops are previously created, and much faster to get at the moment that are needed. Every time the bus routes are updated on the islands we can rerun this task to update the schedule that we have stored.

The setup is now complete. To find the dashboard of a certain display, we need to add the display identifier after the dashboard URL.

Main page. When we access the dashboard URL, we are presented with the main page of the dashboard, which contains analytics for that display environment. The background of the main page is composed of a map of the screen localization.

When we first render the dashboard, the data that is needed for the system to run is all loaded. This happens so we avoid repeated requests, once that the data that is used for all the other pages is the same as the one that is shown on the main page, only on the main page is summarized. Besides that, this way, the data is always synchronized and updated when necessary, and we do not have to wait for the requests to end when we change pages.

The main page is composed of a few different widgets, with different types of information. In the next sections, we will describe each one of them and the different variations they can take depending on the data.

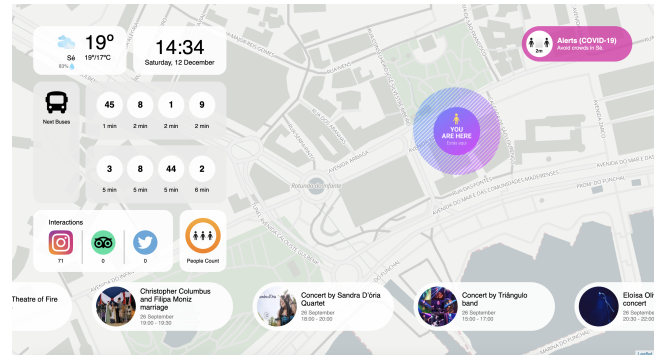


Figure 3: Dashboard's main page

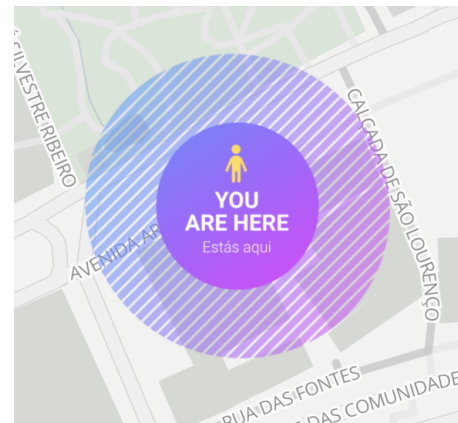


Figure 4: Public display position

Position marker. The marker represented on figure 4 sign the exact place where the public display is located, which means, where we are on the map. The coordinates used for this position are the ones we introduced on the CSV.

Meteorology. The first widget on the top left corner is meteorology. To populate this widget we are using the external endpoints, where we can get the meteorology for a certain location name and for a certain date range. It is refreshed every hour.

We have several meteorology related info on this widget. Looking at figure 5, on number 1 we can find the current temperature in Celsius degrees, which is updated every hour. On number 2 we have the maximum and minimum temperature, respectively, for the day. On 3 we have the current humidity, which is also updated every hour. On 4 is the location name

where the display is at. On number 5 we can find a small image representative of the climate right now. This can take several states represented in figure 6.

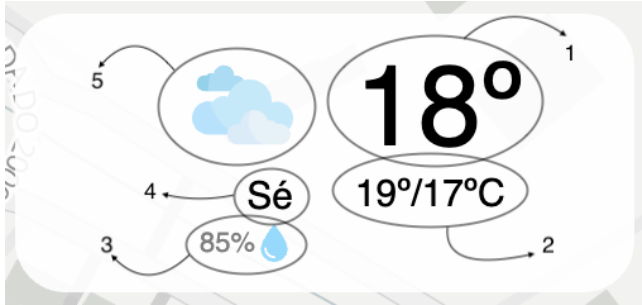


Figure 5: Meteorology widget

The icon on (a) represents the night and is shown from 8 pm to 6 am. During the day, from 6 am to 8 pm, we have four different options. If the 'rain parameter' that is returned by the endpoint is more than zero, (b) is shown. If the 'cloud parameter' is more than 50 we show (c) and if it's between 25 and 50 we show (d). If none of the above is true, (e) is shown.

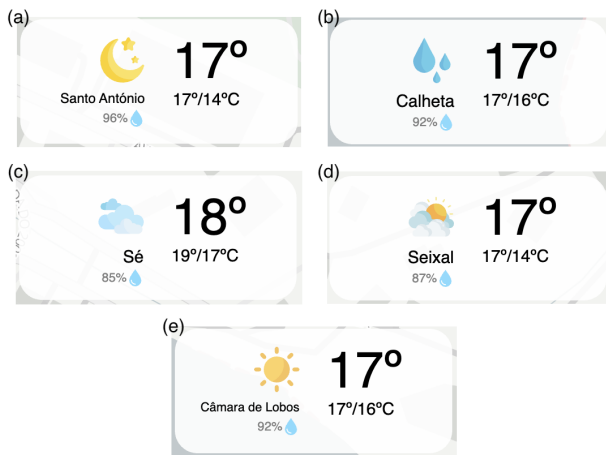


Figure 6: Meteorology icon states

Time and date. The time and date widget simply showcases the current time and date for the timezone the public display is in.

Next buses. Next buses displays how much time is left for the next buses that are linked to this display to pass. Each display has links to the nearest bus stops, and those bus stops are part of different bus routes. On this widget, we can find eight circles, that represent the next eight buses that are going to stop around this public display. Inside the circle is

their route number - so it is possible to identify them - and under it is how much time is left for the next bus from that route.

The information used to populate this widget is stored in the database, as previously explained. All the buses for a day are loaded and stored at the front end and refreshed at the beginning of every day.

The next buses are updated every minute, so the buses that have a time left less than zero, which means they already passed, disappear, the other ones are updated and new routes appear at the end.

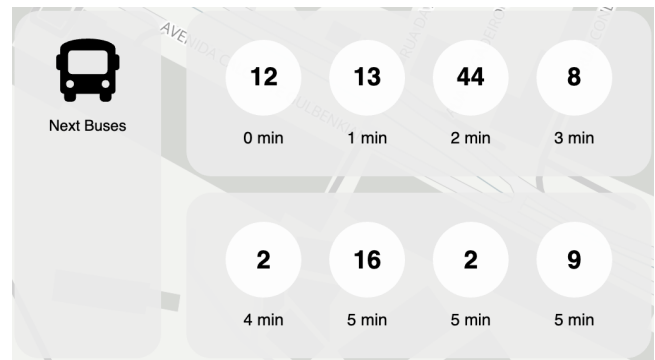


Figure 7: Next buses widget

Interactions. On interactions widget, we can find three numbers related to the interactions of the people with the Madeira Islands in social media. From left to right, we first have the number of Instagram posts related to Madeira that were published today, the number of TripAdvisor reviews of places in the islands, and lastly the number of tweets.

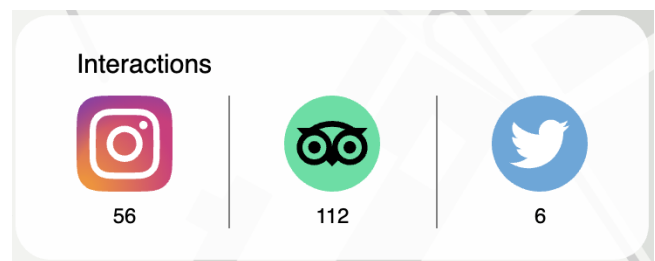


Figure 8: Interactions widget

These values are referent to a day and are refreshed every hour to make sure that are as much up to date as possible.

People Count. People count is the last widget in the vertical structure. This widget showcases the number of people that are around the public display at a given time.

To populate it we are using the external endpoints, which returns the number of the people around a certain router (the public display closest router), at every hour of the day.

This information is refreshed hourly.

Depending on the number of people around, the widget takes different forms. If we have less than 50 people the widget is green (a). If we have more than 50 but less than 200, the widget changes to orange (b). If there are more than 200 people around, the widget is red, as in (c).



Figure 9: People count states

To make this widget more accurate, some further work could be done. The calculations for how many people is too many people could be done taking into account some other factors regarding the place, such as the area, if it is indoors/outdoors, etc.

Events. At the bottom of main page we have a carousel always moving with the events that are happening in Madeira this month. The data for this is consumed from the external endpoints. Although the information is for a month, it is refreshed daily just to make sure that is updated and there are not any new events.

Each card, illustrated on figure 10, represents an event and is part of the carousel. On 1 we have the name of the event, which is truncated if it has more than 70 characters. On number 2 it is the date of the beginning and finishing, if different, of the event. On 3 we have the beginning and finishing hours. On 4 it is an image representative of the event, that also comes from the endpoint.



Figure 10: Events widget card

Other pages. Besides the main page, the system also has three more pages with more detailed information. As previously explained, all the data necessary for the system to run is loaded at the beginning and the same data that is used for

the widgets is also used for the pages. The difference relays in the detail of the information presented, the one on the pages being much more extensive. Being the data the same means that is consumed from the same place that the widgets and is updated at the same time.

Right now we have three more pages: next buses page, events page, and people count page. As the names suggest, they extend the information that the widgets with the same name present.

The idea is to add more pages in the future, with different kinds of information. That would be done easily, without threatening the proper functioning of the dashboard.

Next buses page. On the next buses page, illustrated on figure 11, we have the same information that we have on the widget: the number of the route of the bus and the time that is left for that same bus. Besides that, we also have information about the destination of that bus and the time that the bus is supposed to arrive at the stop we are at. We can also see the name of the public display that we are consulting and where we will hop on the bus.

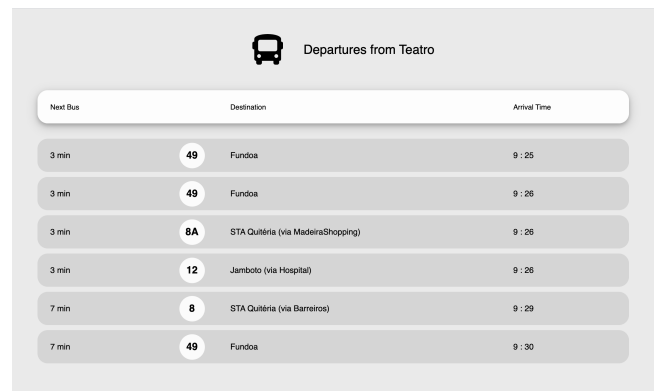


Figure 11: Next buses page

Events page. The events page consists of a carousel that moves every few seconds and shows detailed information about the several events that are happening this month in the Madeira Islands. Each carousel item has the size of the screen, as we can see in figure 12.

At the top, we have an image relevant to the event and the respective title. On the right side of the page, we have the starting time and date, the ending time and date, and the location where is going to take place. Lastly, we can also have a small description of the event and one or more categories that were assigned to it.

People count page. The people count page's goal is to show off where are the most crowded places in the neighborhood. Having this information available is not only useful in the

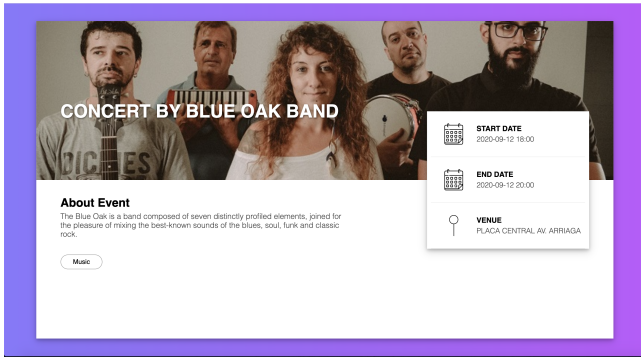


Figure 12: Events page item

pandemic situation we are living in at the time of the writing of this work, but will also allow us in the future to know what places are more crowded.

This is the only page that has different information comparing the page and the widget. While in the widget we show data about the number of people around the nearest router of the public display, we here present information from several routers that are within a radius of some meters of the public display. Each bubble is centered on one of those routers, and its size and color depend on how many people are passing that router, as stated in the legend. The diameter of the bubbles is calculated using a logarithmic function. In the bottom bar, we can also find the names of the places where the routers are, which are the center of each bubble.

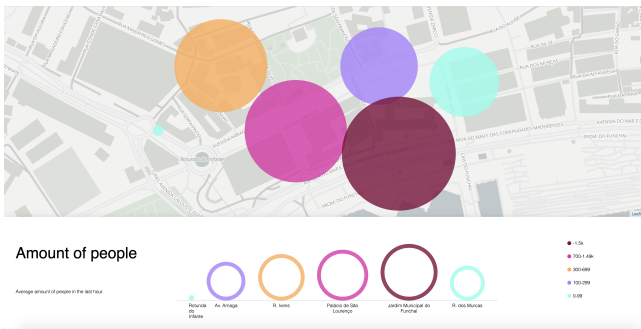


Figure 13: People count page

This bubble map chart was built with the Leaflet library, as the main page. As the widget, the accuracy of this page could be improved with some more information about the places, which would allow us to understand how crowded it is, but that was not in the scope of this work.

Notification. Besides the widgets and the pages, we have one more component: the notification. The notification is a component with a structure as represented in figure 14, that appears at the top right corner of the dashboard and can

highlight varied information. It appears every other minute, lasting one on the screen.



Figure 14: Notification

As of now, the purpose of the notification is to draw attention to viewers to respect the social distancing, showing off what are the most crowded places at the moment. That way, people can keep away from those places, avoiding clusters.

As explained on the people count page subsection, we collect information about the number of people from several routers that are within a radius of some meters of the public display. If there are two or more places that count more than 200 people around, the notification shows the names of the two that are more crowded. If only one of the routers captures more than 200 people, the notification only showcases that one place. If none of the routers detect more than 200 people around, the notification does not show up.

The notification is independent of the other pages, can appear in front of any of them.

Personalization. When we access the dashboard URL, with the public display identification, we are presented with the main page. The way we navigate to the other ones is the key to the personalization component of the dashboard.

One of the requirements of this work was for the dashboard to be personalizable through the mobile phones of passers-by. The authentication of the user and access to the communication with the public display is not on the scope of this work, so in the project and description we assume that the user is authorized to communicate its preferences to the display.

The personalization starts with the viewer having on its mobile phone an application that allows choosing what kind of information he/she wants to see in the public display that he/she has around. The development of that application is not in the scope of this work. When the viewer has a display in his/her surroundings, he/she can send, through that application, what he/she wants to see in more detail. The preferences of the viewers are saved.

Every minute, some criteria are evaluated in order to understand which page should be shown next. For the time being, we have two criteria: how many viewers prefer to see each page, and how many rounds there have been since the last time each page was shown. There are some exceptions

but briefly, this means that if a user expresses a preference on two different pages, the one that is shown is the one that was shown lastly more rounds ago. If no user expresses a preference, the main page is shown. The decision-making process is explained in detail in the next section.

Architecture

To develop the project we used a layered architecture pattern. This pattern, as the name says, is organized into horizontal layers and each layer performs a specific role and responsibility within the application (in this case, presentation, business, database) [12], as we can see in figure 15. This pattern supports the incremental methodology and creates abstraction and isolation around each layer.

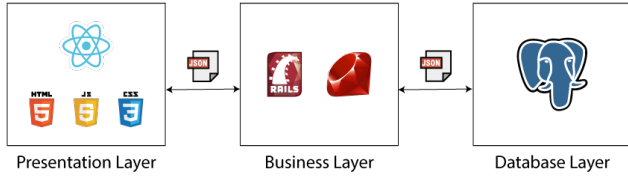


Figure 15: System architecture

Our application has three layers: presentation layer, business layer, and database layer. They communicate with each other through JSON files. A JSON file is a file that stores simple data structures and is a standard data interchange format.

Database layer. The database layer is responsible for managing and storing the local data that is used in the system, the one that is not directly consumed from the previously presented external endpoints at runtime. It was developed in PostgreSQL.

The database will be populated when we do the setup through tasks that will process data from CSV files or the endpoints available, creating its instances and relationships between them.

Business layer. This layer is responsible for, not only all the business logic that will be necessary but also for making the connection between the presentation layer and the database, having an API role. The technology used to develop this layer was Ruby and Ruby on Rails.

It is on the business layer that we do the requests to the external endpoints and processes their responses. The personalization is also taken care of by the business layer. Assuming that the viewer is authorized to interact with the public display, the viewer can send its preferences to the system through a WebSocket. The WebSocket was integrated with this Rails application using Action Cable.

The viewer preferences are stored in the browser sessionStorage. We are using sessionStorage because the viewers'

preferences are irrelevant some time after they were done since the viewers are expected to be in movement and not to be waiting for the public display response. Besides the viewers' preferences, we also store how many rounds ago each page was shown last.

To send preferences to the system you need to broadcast the requests. The function broadcasting allows streaming the message to the channel indicated.

Whenever a message comes in through the channel the received method is called. In the received method we format and add the new preferences to the sessionStorage.

Every minute we get was is stored in the sessionStorage and send it to the decision-maker, implemented in Ruby on Rails.

Decision-maker. What page should we show next? This represents a multiple-criteria evaluation problem since our question has a finite number of alternatives, explicitly known at the beginning of the solution process. What we aim for is to build a decision-maker that finds the best alternative.

Since this is a single-dimensional problem, we opted to use a simple weighted sum model. The weighted sum model (WSM) states that, if there are m alternatives and n criteria then, the best alternative, is the one that satisfies (in the maximization case) the following expression[13]:

$$A_{WSM-score}^* = \max_i \sum_{j=1}^n a_{ij}w_j, \quad \text{for } i = 1, 2, 3, \dots, m.$$

where: $A_{WSM-score}^*$ is the WSM score of the best alternative, n is the number of decision criteria, a_{ij} is the actual value of the i -th alternative in terms of the j -th criterion, and w_j is the weight of importance of the j -th criterion.

Our algorithm starts by checking if there are any preferences that were got from the sessionStorage and sent to the decision-maker. If not, the main page is shown and the evaluation stops. If there are, we apply the weighted sum model to find the best option. To apply the WSM to our concrete problem, we generate a matrix similar to the following one, with the data we got from sessionStorage.

Page	Number of preferences (0.50)	How many rounds ago (0.50)
Buses	10	2
Events	3	8
People count	8	5

Table 1: Decision matrix for the WSM

Each criterion has a weight, in our case the two criteria weight 0.50. We multiply each cell by the respective weight and then add every value of each row. When the formula is

applied to the previous data, the scores of the three alternatives are:

$$Buses_{WSM-score} = 10 \times 0.50 + 2 \times 0.50 = 6$$

$$Events_{WSM-score} = 3 \times 0.50 + 8 \times 0.50 = 5.5$$

$$People_{WSM-score} = 8 \times 0.50 + 5 \times 0.50 = 6.5$$

If the biggest value has zero votes (there are not any viewer that signed it as a preference), meaning that only won because it is was not shown for a long time, or if the biggest value was shown one round ago, which means that was the last page shown, we show the main page. Otherwise, we show the page with the biggest WSM-score.

Every time a page is shown the count of how many rounds ago the page appeared lastly is changed to one, while we add one to the count of the other pages. After ten minutes of a particular preferences request, that request is deleted, since we expect the viewer who done the request to not be in the surroundings of the public display anymore. That is why the preferences are stored with a time stamp.

Presentation layer. The presentation layer corresponds to the front-end of the platform. It is responsible for creating and displaying the user interface and handling user interaction. Data that is shown on the front-end is fetched from the business layer.

This layer was developed using React, HTML, CSS and JavaScript. React is a Javascript library for building user interfaces. It makes possible to design simple views, using HTML and CSS, and React efficiently updates and renders the right components when data changes.

For maps, we used Leaflet. Leaflet is a JavaScript library for creating interactive maps. It works efficiently across all major desktop and mobile platforms and can be extended with lots of plugins.

Each of the widgets previously presented is a React component. This way, we can add and remove components without compromising the good functioning of the dashboard. The main page, and the other pages, are also each one a component. The page that is selected to be shown, is the one that is rendered.

4 EVALUATION

After we have completed the development of the functional prototype, we started to think in a better way to find if our system met the needs of users - people can gather useful information at the same time that they feel that the system is part of the environment.

Methodology

This evaluation has as the main function to evaluate the best and worst aspects of the functional prototype and to check that a standard was upheld, a process known as summative

evaluation [11]. Users are presented with a list of tasks, and their performance is evaluated through quantitative measures: the time it takes the user to do the task, the number of errors made (if any), and the level of satisfaction while doing such tasks.

To perform these tests we used 10 users, with no special age, background, or technological literacy filtering because we used a sampling technique called Convenience Sampling, which means that our users were selected because of their convenient accessibility and proximity to us.

Protocol

To maintain the consistency of the tests, we developed a protocol to ensure that all users went through the same process.

The protocol was divided into the following parts:

- (1) A profiling survey was used to collect key demographic information about our users, such as gender, the range of age, education, birth country and if the users had a smartphone.
- (2) A contextualization of our project and evaluation method. We followed a prepared script to ensure that we did not forget to explain anything and to ensure every user had the same information.
- (3) The execution of a list of 5 tasks to be performed one at a time and without any help. The users were asked to inform us when the task started to start the timer. In order to provide more comfort to users, they were informed that what was being tested was the system and not them.
- (4) A second survey to understand the users' level of satisfaction while doing the tasks. For this we used the System Usability Scale (SUS) - a simple, ten-item scale that gives a global view of subjective assessments of usability [1] about the user experience with our system. We followed the guidelines established by Brooke [1]: each question had a degree of disagreement or agreement, with a range from Strongly Disagree (1) to Strongly Agree (5) respectively, from which the user could choose. Users were asked to answer each question with their true opinion, but we recommend them to not think too much about it, and if undecided to pick the middle score of the presented scale.
- (5) Final thoughts and offer of cookies to tested users as thank.

For the tasks to be done by the users, we tried to choose some that covered the maximum points of the system to make sure that we would discover the maximum possible problems.

Results

From the results of the usability tests and the score of 83.75 points following the guidelines of John Brooke [1], we can assume that our system has a good acceptance and usability rate.

Some problems and the targets for improvement were raised in carrying out the tasks. Social networks should be identified not only with the logo but also the name, the various carousels of the system should have a lower speed so that users who read more slowly can do so. On the events page, since each event has a lot of information in text, some design options could be rethought and prioritized so the user can read the information faster. On the bus page, the number that identifies each bus should be clearer for the user.

Two points of improvement were detected during the tests, but not directly related to the various tasks. The first was the language, which we realized that in a real system you should have the option of the language of the country where the system is being used. The second was related to the font size of the whole system, which should be a little bigger so that users do not have reading difficulties.

After the tests are finished, we can say that the system ends up meeting our initial objectives and ends up corresponding to the needs of our target audience.

5 CONCLUSION

After developing and testing the functional prototype, we can conclude that, in general, the goals set out have been achieved. The dashboard presents the data that it should, allowing users to interact with it through their mobile phones and choose the page they want to see next. The system, when there are conflicting requests by several viewers, chooses the one that is the best option taking into account the criteria chosen. This does not question the user's privacy or disclose its entity.

Besides the small improvements already mentioned as a consequence of the tests of the system, there are also other features that could be added and some improvements that could be made to the already existing ones.

Starting with improvements to the features that we already have, we could add data to our data sets about the places where the routers are implanted so, that away, we could take those into consideration when calculating how many people is too many people in a place.

Besides that, the system could be deployed in other places of the world other than Madeira without any change if the endpoints used for data consumption were the same. If not, the system can be adapted in order to work for other cities.

Regarding the decision-maker machine, more criteria can be added easily, or change of the criteria weight, in order

to refine the page chosen, trying more and more that the chosen page is the ideal one.

More widgets can always be easily added to the dashboard, it is only necessary to develop the new widget, collect the necessary data and add it to the dashboard. The same for the pages, in which case it would be necessary to add the page to the decision machine.

REFERENCES

- [1] John Brooke. 1996. SUS: A quick and dirty usability scale. *Usability Evaluation In Industry* 15, 8 (1996), 207–212.
- [2] Nigel Davies, Marc Langheinrich, Rui Jose, and Albrecht Schmidt. 2012. Open Display Networks: Towards A New Communication Media for the 21st Century. *IEEE Computer* 45, 1 (2012).
- [3] Niklas Elmqvist and Pourang Irani. 2013. Ubiquitous analytics: Interacting with big data anywhere, anytime. *Computer* 46, 4 (2013), 86–89. <https://doi.org/10.1109/MC.2013.147>
- [4] Elaine M Huang, Anna Koster, and Jan Borchers. [n.d.]. Overcoming Assumptions and Uncovering Practices - When Does the Public Really Look at Public Displays? ([n. d.]), 228–243.
- [5] Rui José, Nuno Otero, and Jorge C.S. Cardoso. 2014. Dimensions of Situatedness for Digital Public Displays. *Advances in Human-Computer Interaction* 2014, December (2014). <https://doi.org/10.1155/2014/474652>
- [6] Heinz Kautz, Karlheinz; Kuhlenkamp, Karin; Züllighoven. 1992. What is Prototyping Model? *Northwind Publications, Ltd.* (1992). <http://searchcio.techtarget.com/definition/Prototyping-Model>
- [7] Daniel Michelis and Jörg Müller. 2011. The audience funnel: Observations of gesture based interaction with multiple Large Displays in a City Center. *International Journal of Human-Computer Interaction* 27, 6 (2011), 562–579. <https://doi.org/10.1080/10447318.2011.555299>
- [8] José Miguel and Santos Ribeiro. 2016. Human Mobility Tracking through Passive Wi-Fi: a case study of Madeira Island. (2016).
- [9] Mateusz Mikusz, Kenny Tsu Wei Choo, Rajesh Krishna Balan, Nigel Davies, and Youngki Lee. 2019. New Challenges in Display-Saturated Environments. *IEEE Pervasive Computing* 18, 2 (2019), 67–75. <https://doi.org/10.1109/MPRV.2019.2906992>
- [10] Jörg Müller, Florian Alt, Albrecht Schmidt, and Daniel Michelis. 2010. Requirements and design space for interactive public displays. *MM'10 - Proceedings of the ACM Multimedia 2010 International Conference Figure 1* (2010), 1285–1294. <https://doi.org/10.1145/1873951.1874203>
- [11] Helen Preece, Jenny; Rogers, Yvonne; Sharp. 2011. *Interaction Design: Beyond Human-Computer Interaction*.
- [12] Mark Richards. 2015. *Software Architecture Patterns*. O'Reilly Media, Inc.
- [13] E Triantaphyllou. 2000. MCDM Methods: A Comparative Study. (2000).
- [14] Mark Weiser. 1999. The computer for the 21 st century. *ACM SIGMOBILE Mobile Computing and Communications Review* 3, 3 (1999), 3–11. <https://doi.org/10.1145/329124.329126>