

Information System for Animalife

Tiago Neto

Instituto Superior Técnico

Lisboa

tiago.neves.neto@gmail.com

ABSTRACT

Animalife uses a system composed of multiple Excel sheets, being difficult to manage and use because there is the possibility of duplicate information on different sheets.

From the point of view of a non-profit organization, we observe that in this case, “time” is a resource of even higher value because for this association to remain operational and capable of handling aid applications received, we must consider that their main collaborators are volunteers, so allowing them to perform as many operations in the time they have, will enable helping more families and animals.

Animalife realized that the usage of their current system was not the best option, the need for an information management alternative arose.

Considering the nature of the project and of the organization, it is one of our main requirements that the cost of development of this system is null. To resolve this issue arose three development options. Two of the three options met the necessary requirements and were then classified so as viable.

The opted solution was the most innovative one.

This solution presents the usage of Google Drive as a database. It presents the benefit of easy data management for a “common” user, who does not hold advanced technological knowledge.

Finally, in this document, the results obtained by the evaluation of the system users are presented, and possible features to add to the system in newer versions are listed, as well as important information to those who are considering a similar approach.

Author Keywords

Time, Volunteer, System, Information management, Google Drive.

1. INTRODUCTION

Animalife is a non-profit association of Awareness and Social and Environmental Support, which was established in October 2011. It is nationwide, and is headquartered in Lisbon and has the support of more than five decades of volunteers who dedicate part of their time to this cause. Animalife's fundamental objectives are the promotion of citizenship, environmental protection, public health and the protection of disadvantaged people.

Contributing to support the institutions which are intended to accommodate the stray animals, promote their

vaccinations, deworming, sterilization and consequent control of the overpopulation of dogs and cats, particularly in terms of organization and management of these institutions. Conduct and support initiatives aimed at improving the quality of life for families in need, through elimination of food shortages, providing help to pets that are in their care, thus preventing the abandonment of animals and over-population in kennels and Associations hostels.

Animalife to achieve its goals, can promote and develop initiatives to sensitize the population, including boosting and supporting other associations in organizing events and initiatives related to the purpose of the association.

Animalife is the manager and organizer of the National Campaign for Food Collection Animal - Animal Outreach Bank - which aims to provide all the associations and Families at risk a direct benefit of helping the animals that are in their care.

1.1. The problem and its motivation

Currently all its data about families and animals is stored in Excel files. Information on these can be spread over several documents instead of being consolidated all in one place. This makes the operations become longer and more exhaustive than needed. This problem coupled with the fact that for various locations, the data is collected and recorded in various ways contributes to an outdated system, which is time consuming and messy.

The entire process of registering a new pet or family at present is done through Excel sheets, not the best solution not the most practical for this purpose.

1.2. Objectives and constraints

To solve the problems mentioned in the previous point, the system to be developed must meet the following objectives

- Increase the efficiency of users
- Reduce the complexity of the processes
- Consolidate the storage of information
- Make a good organization of data

So, it was necessary to carry out an organization analysis to understand how they proceed to the execution of the tasks they perform. After this analysis we can understand the improvement of space that exists for each one, simplifying the task whenever possible.

The last two goals stem from flaws identified with registration of information, in the system that currently use. So, not wanting to make the same mistake and based on the analysis above, the information is organized according to its relevance, ensuring that there is no duplicate data.

1.3. Results

For the implementation of this system the following technologies were used:

- HTML
- PHP
- JavaScript

In addition to these technologies it was also used Google Drive as data storage, where the data is inserted into spreadsheets named. With the usage of these technologies it was possible to implement a system capable of performing the management of families and animals. Additionally, it's also possible to manage volunteers.

Management is the entire process of creation, modification and removal of data from each of these modules, for example, in the case of families' management you can:

- Add a family
- Add a new family member
- Add a new pet
- Conduct a technical assessment
- Develop a recognition assessment
- Remove a family member
- Remove a pet

The same goes for the management of animals and volunteers who have similar functionalities although according to the environment where they are inserted.

All these features have been tested and approved by the association and it was also carried out at a usability a system assessment based on the System Usability Scale model.

2. ANALYSIS OF THE PROBLEM

From the problem identified in section 1.1, comes the need for computerization of different documents so that, regardless of the location, there is a pattern, thereby facilitating the exchange of information when needed and for easier storage of information related to users.

The new system to develop will improve the company's organization of documents and information contained in them and contribute to the optimization of its processes, making them faster and more agile.

Thus, considering that the company is mostly staffed by volunteers and depends on the time that they provide, this optimization will be of high value because it will allow them to make better management of their time, making it possible to execute more tasks in a quicker and more efficient manner.

2.1. Solutions

This section will describe the solutions considered for the implementation of this project and finally it will also describe the solution chosen explaining the reason.

2.2. Solution A

This solution uses PHP, MySQL and HTML.

Thus, to design a prototype based on these supports there was a need to review some concepts of MySQL and HTML, as well as the improvement of knowledge about PHP.

At the end the developed prototype presented the basic features of the project.

2.3. Solution B

This solution is supported on MongoDB, Express, Angularjs and NodeJS.

MongoDB is a Database Management System (DBMS) of the type NoSQL (Not only Structure Query Language). It is Open Source and developed in C ++ designed for storing large amounts of data (documents) with JSON format.

Express is a framework for Node.js designed for web development. AngularJS was made to extend HTML adding new attributes. It is usually used for single page applications.

Node.js was developed in JavaScript, it is an asynchronous library triggered by events. It is designed to build scalable web applications. A solution that integrates these four components is usually called a MEAN stack application (MongoDB, Express Angularjs and Node.js).

This solution showed some advantages in applying terms of maintenance of processes compared with the above solution (A).

Since it uses the JavaScript programming language, to develop a prototype it was essential to increase the knowledge of JavaScript and MongoDB[4].

2.4. Solution C

This option is based on Google API Sheets and Google Drive API.

There was also the need to study the Google Sheets API.

For the implementation of the project we used version 3 of Google Sheets, as it was the version with more support. The support that exists on the latest version (v4) is still very low since it was launched in the market recently.

We choose to use a PHP framework, called MINI. This framework is minimalist, containing only the essentials for an MVC software architecture standard. Because of this, it is easy to learn and use.

To further facilitate application development, it was used a PHP library that makes interaction with version 3 of Google Sheets.

The created spreadsheets in Google Drive will act as our database and is where all these data will be maintained.

The ParsleyJS library is used to validate the existing forms in the application. This already contains its own tools for validating email, numbers, strings, maximum size / minimum of one field or to check that a given value is

within a range with an upper and lower limit already set, among other possible verifications. The next section explains why the creation and usage of our own forms instead of the usage of Google forms.

This solution has undergone several changes. First it started as a console application and was later converted a web application.

The console application, created according to Google tutorials, served as initial testing of the application, so that in a first phase, it was possible to see what the limitations and constraints of using this tool in this project.

There been several iterations of the "layout" of this application, initially it began as an application that ran through the console until finally we opted for a "Dashboard" style. This transition in the appearance was made to allow the users of this association, to view and manage all data related to their stakeholders easily.

2.5. Comparison

To decide on the best solution to address, it is necessary to understand the advantages and disadvantages of each.

Solutions A and C have many similarities, their unique and crucial difference is the system used for data management.

In the case of the solution A there is the usage of a Database Management System (DBMS), MySQL to manage all data necessary for the execution of the desired operations.

On another hand solution C, does not use a DBMS, it uses Google Drive as the database of our system. Compared with the solution mentioned earlier, typical MySQL tables are then replaced in this solution by spreadsheets and it is here where all data is maintained and managed.

Regarding solution B, as it uses a database that is paid, MongoDB, and this project is being developed for a non-profit organization one of the main requirements is the cost of technology to adopt, so this solution was then abandoned because it does not meet this requirement. So, the choice relapsed on one of the two viable solutions, the usage of MySQL or Google Drive.

Both technologies are free, but Google Drive has only currently free 15GB, while MySQL has a database that is only limited by Windows itself, varying in size depending on the distribution of the Operating System (OS) used.

MySQL is currently one of the most popular systems used for the management of databases and has over 10 million installations worldwide. Advantages:

- Easy to install and has many third-party tools;
- It is open source which means that there is a large community which can be called upon if necessary;

Disadvantages:

- The degree of performance tends to decrease with the level of scalability;

- Its functionality tends to depend a great deal on Addons which may involve a significant loss of time in the search for an application.

Google Drive has 240 million users, and is a file system with synchronization service developed by Google. The use of Google Drive as a database has some advantages:

- No knowledge of SQL or NoSQL is required.
- It is easy to understand for a user without technical knowledge.
- • Allows sharing of files between users.
- Has versioning built-in.
- Allows access to files offline.
- Synchronization across multiple platforms (Windows, MacOS, Android, iOS).

Limitations:

- Document Storage up to 50MB (in Google Docs format)
- Storage spreadsheets with up to 2 million cells.
- Presentations of storage with a maximum of 100MB (Google Slides format)
- Storage of any other type of file up to 5TB.

There is another drawback when using Google Drive with database compared with MySQL in terms of performance, it takes longer to perform the same tasks.

In the case of MySQL usually the database is housed at the same site as the server and the communication time between PHP and MySQL is virtually nil.

By using Google Drive that is no longer the case, because to be able to save the submitted files or changes made to the same it is necessary establish a connection with the Google Drive server and only after that we can proceed to executing the operations. The connection time is volatile so depending on the server connection quality and distance from to the Google Servers.

As it was expected the Google Sheets is not the best in terms of performance, that is because it was not designed and developed with the purpose and intention of being used as a database, so compared to a DBMS, it is natural that there is this difference in results.

The choice of using Google Drive as a database for this application relapsed in that it is a technological innovation, but also on the advantages mentioned above. As this system is to be used by a non-profit association, the use of this technology allows the database maintenance without requiring specific knowledge of SQL.

2.6. Adopted Solution

For the development of the system solution C was chosen.

In terms of the method of development we opted for an agile approach[2], adopting the characteristics of this method of development and following the its best practices.

In terms of technology, and according to customer

preference for free technologies it will be used HTML5, JavaScript, PHP and Google Drive.

The choice of these technologies considered the level of complexity of the project. This new system will have a certain level of complexity and some dimension, but these attributes do not represent a problem because they do not exceed the limits of Google Drive.

The user interface, will provide the association and its employees a visual component to the inclusion of data on their users without requiring any knowledge of programming languages.

Thus, the user interface will be an advantage because it allows the volunteers to make the filling of data a continuous and immediate way, without having to worry about what document to access for this purpose as they used to do.

This user interface provides thereby a much faster way to enter all the necessary data. The data will be carried to its correct spreadsheet.

This new system will also have the two modules referenced above, and the module that will allow the management of volunteers.

Present in this section is an analysis of user needs, resulting in user stories of users and a use case diagram that reflects the needs of users and their interactions with the system.

2.7. Functional Requirements

An important step to software development is identifying the functional requirements of a system.

The functional requirements described below were obtained through the user histories.

A user story is a set of one or more sentences that describe what the user of a system does or need to do in their day to day considering the needs of their work.

To better understand the needs of the users in the system to be developed, it is crucial to understand where the entity in question belongs to in its area of operation, as well as the current needs of a user and to consider their future needs.

After having this perception, it will be possible to map these needs for the future information system.

According to the referred in the preceding paragraphs we have obtained the following user stories:

1. As user, I want to add a database to the family to have the recording information of this;
2. As user, I want to make the removal of a family to be able to proceed to update the list of families in need of support;
3. As user, I want to be able to add new family constituents (people or animals), to a family, to be able to keep the data up to date.
4. As user, I want to remove family constituents (people or animals) to keep the data updated.
5. As user, I want to make a financial assessment on the family, to determine the type of financial situation you are in.

6. As user, I want to perform recognition assessment to determine the conditions of a family.
7. As user, I want to prepare technical reviews of a family, to determine the type of support you need.
8. As user, I want to be able to manage approve volunteers to increase the number of volunteers to use the system.
9. As user, I want to edit a volunteer, to be able to assign different permissions to a registered volunteer.
10. As user, I want to be able to remove volunteers, to restrict access to the system when they finish their volunteering.
11. As user, I can change my personal data to keep this information up to date.

The use case diagram allows an interpretation of the system as well as a graphical representation of the stakeholders and how these relate to each other. The use cases are visually represented in the following figure.

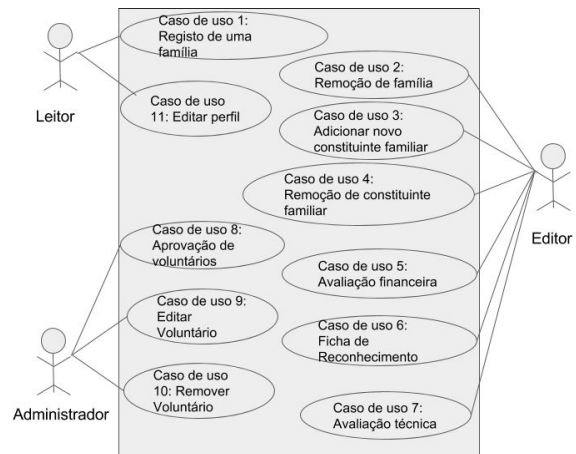


Figure 1- Use case diagram

2.8. Solution Architecture

The following diagram, shows the standard of architecture, Model-View-Controller, which will be used in this approach.

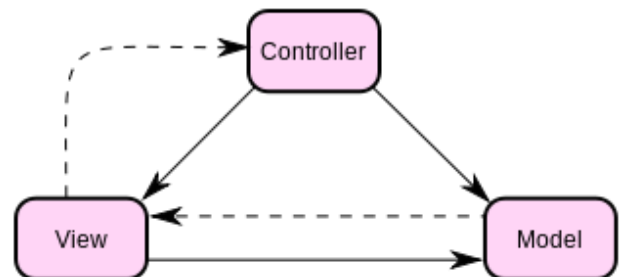


Figure 2- Model-view-controller

The model shown in the figure above, is where all the storage and manipulation of data is done, which means that it interacts with the database. It's also responsible for all logic and definition of system rules.

The controller in the same figure (figure 2), is what allows the user to interact with the system. Thus, in this case it corresponds to the user interface.

The user interface corresponds to the entire execution of operations by the user. When necessary the controller can interact with the model to update the data and it can also communicate with the view whenever it needs to know that there is a change, so that, that change can be displayed to the user. It should be noted that the user interface is a restricted access area, access is allowed only to registered users.

The view in this figure 2, is responsible for the way that data is made available and how it is shown to the user, these may be presented in different forms (text, graphics). When there are changes in the model, the view updates its presentation so that the data shown is in accordance with the current model.

2.9. Used Technology

In this section the technologies used in development of the system, will be described and analysed.

2.9.1. HTML

Hypertext Markup Language is a content development language for the "World Wide Web" and is a very important element of the Internet. Over the years new versions have been launched and currently, the latest version is HTML5. Most current browsers already have native support for HTML5.

HTML5 benefits:

- Compatible among the major browsers, such as Firefox, Google Chrome, Opera, Safari and Internet Explorer. This does not mean that all these supports all new HTML5 elements but at least are able to read the doctype;
- Good performance on most devices, PCs, mobile devices, tablets;
- Support for video integration without the use of plugins. Disadvantages:
- Older or outdated Internet browsers may not support HTML5.

2.9.2. JavaScript

JavaScript is a language used to typically extend the functionality of websites. JavaScript appeared in 1995, first it had the name of Mocha and since then its name has undergone some changes until the current name.

It is a simple and versatile language, it is currently one of the languages most commonly used in client-side programming, it's also being used on the server side due to environments such as Node.js

Advantages:

- Being executed on the client side uses resources customer, thus saving the web server and its resources;

- It is a relatively easy language to learn. Use model Document Object Model (DOM), which provides pre-written functions for the elements of the pages which facilitates the development of scripts;
- Does not depend on the server and running on the client side that allows quick execution of scripts;
- Allows data validation, for example, in the case of customers or login registration, it stops invalid request and, so it prevents from overloading.

Disadvantages:

- You can write malicious JavaScript code that when attached to web pages, is executed immediately on the client side and allows you to explore the system. Although there is a restriction level, defined by the web standards, in browsers malicious code can still run;
- JavaScript rendering varies according to the different layout engines and can produce different JavaScript results which leads to inconsistencies in terms of features and interfaces. While the latest versions of JavaScript and rendering were directed to a universal standard, there are still quite a few variations.

2.9.3. PHP

PHP came in 1995 created by Rasmus Lerdorf, originally called Personal Home Page, now known as Hypertext Preprocessor. It is mostly used on the server side and for applications that generate dynamic content for the World Wide Web.

It can be used in conjunction with HTML and it is a very similar language in terms of data types, syntax, and functions to the C and C++ languages.

Advantages:

- Allows interaction with databases, eg MySQL;
- The PHP installation is possible to do on most operating systems, free of costs;
- Open source which means making it easier to support and maintain.

Disadvantages:

- The code is interpreted on the server side by PHP module, which can lead to a higher response time;
- Being open source, anyone has access to the source code and can take advantage of this, bringing security problems;
- Not being a modular language is not suitable for large applications it is difficult to maintain

2.9.4. Google Drive

Considering that for this project we are using external resources such as Google Drive and Google Sheets, we are restricted by the limitations of these tools. When it comes to Google drive, there is a limitation of storage space, which is currently 15GB, among others, all mentioned in section 2.5.

Regarding the spreadsheets that we will use as tables of our database we can observe the following limitations.

Limitations of spreadsheets:

- 400,000 cells per sheet;
- 256 columns;
- 40,000 cells containing formulas;
- 200 Worksheets for each spreadsheet.

Comparing a spreadsheet with a MySQL table and whereas the worksheets were not designed to be used as a database, there is a significant difference in write access and smaller difference in reading access.

These differences become very relevant the larger the size of data to be entered in the worksheet. Which means that the bigger the amount of data that needs to be inserted into the database the longer it will take to execute this operation.

There are also limitations on the use of Google Forms, you cannot create a question that is answered in a matrix form, this is important in our case to fill information about the household and their animals.

Thus, for this case, it was required to create a specific form to be filled by the user. To maintain constant visual look over the application we chose to use only forms themselves. Unfortunately, in making this commitment we lose the ability to use the built-in analytical capabilities in Google forms.

2.9.5. MINI

MINI is a whole framework developed in PHP, it is extremely simple to use and learn and is a minimalistic framework, i.e. it only has the necessary and basic to be able to use an MVC style approach.

Technological requirements of MINI:

- PHP version 5.3.0 or higher.
- Active mod_rewrite on Apache settings.

Features:

- Supports database interactions.
- Allows usage of AJAX .
- Usage of only native PHP code.
- Allows CRUD actions.

MySQL usage is supported by this framework, but it's not defined as requirement in terms of database.

2.9.6. PHP Library

To interact with the spreadsheets, it was decided to use a library that makes interaction with version 3 of Google Sheets thus allowing a faster development.

This library[3] has the function of finding a spreadsheet present in Google Drive, using its name or identifier, returning the spreadsheet itself. After obtaining the spreadsheet it possible to access the information written on the sheet and it is also allowed to add new content or change existing content.

To install this library the usage of Composer is necessary.

2.9.7. ParsleyJS

Parsley is an open source project. It is a plugin developed in JavaScript to validate forms. This plugin has capabilities to perform various validations such as:

- Pattern Validation (E.g., "[a-zA-Z] 4 [-] [a-zA-Z] 4")
- Establishing a maximum/minimum length of a field.
- Validating that two fields have the same value.
- Assess whether a string is alphanumeric.
- By using this plugin, evaluation of fields on a form is done automatically saving time and reducing errors.

2.9.8. CSS

To design page layout CSS was used, this technology allows the specification on how you want the HTML elements to be displayed on the page, and can even create multiple layouts for the same page.

2.9.9. Twitter Bootstrap

Twitter Bootstrap is a tool that contains several elements ranging from CSS components, used for example on the creation of a navigation bar.

It was designed to speed up web development and therefore contains most of the requirements for the it. For the development of this project it was heavily used and can be visible in almost every screen.

2.10. Prototyping

To develop this project, we chose to follow a prototyping method of the evolutionary type.

In this prototyping gender, the goal is to create an initial prototype, which undergoes multiple iterations over time to be refined and becomes more robust.

It begins by implementing the functions for which the requirements are easy to understand and are well defined. After the implementation, these functions are then tested so that later the necessary corrections or improvements can be made.

This technique allows the programmer to add functionalities that may have escaped during the step of collecting requirements.

In this approach, it is considered that a system is never in a "final" state because there is always room for improvement that arises with the use of the product or which is triggered by the change of environment where the product is being used.

This type of prototyping allows for the development of the components of the system according to the level of understanding of the requirements permitting changes to be made as soon as possible.

3. IMPLEMENTATION

In this section, we will address the two most important modules for the implemented system.

The first of these is the volunteer management module.

For a volunteer to have access to the system it is first necessary to submit his registration in the system and later receive the approval of an Administrator.

There are three types of users:

- *Leitor*
- *Editor*
- *Administrador*

There six regions of access:

- *Lisbon*
- *Margem Sul*
- *Oeiras*
- *Porto*
- *Sintra*
- *Tudo*

When performing the approval of a user the administrator must assign the permission level from among the three levels referred above, as well as a user access area.

The access region serves to filter the access that each user has to the information present in the system.

The level of permissions is used to filter access to certain operations such as:

- Removal of people
- Removal of animals
- Removal of families

Therefore, only users with equal privileges to "Editor" or above can perform the previous operations.

The user management menu allows an administrator the following operations:

- Edit a user
- Approve a user
- Remove a user
- Activate an inactive user

Note that when a user is removed he goes from an active state to inactive state, and is therefore possible to return to the active state in a near future.

The family management menu is available to all users, but as mentioned above there are operations restricted to certain users.

It is here all the data management of all families, allowing operation such as:

1. Registration of a new family
2. Assessment of the financial situation of a family
3. Editing family data
4. Removing pets
5. Removing family members
6. Adding recognition records

7. Elaboration of technical support evaluations

Points 1 and 2 can be made from the initial page of family's management screen, the remaining points can only be executed within the screen "Detailed view" of each family accessible through its process number.

When you perform the registration of a family all data relating to the family and to their household and pets is saved in the database. The generation of the process identifier is done automatically based on the region where the family is included.

Through the financial evaluation, a user will make, according to the data collected, an analysis on the family's financial situation.

From this analysis, it is determined whether the family meets or not the required requirements for aid.

Recognition records have the following objectives, among others, document the living conditions of the family, make a follow-up visits to the family to check on them.

The technical support evaluation is where it is determined what type of support a family needs, this phase relates to point number 2.

There is one last option available in the user management menu that is, the recovery of a removed family, because it is possible that a family will need help in the future or even to retrieve a family that was wrongly removed.

4. EVALUATION

Evaluation with users is an important step in the development and implementation of an information system as it is with this step that we ensure that the system meets the expectations of the customer.

It is also essential at this stage to make an evaluation phase with future users of the system, to gather sufficient data to elaborate an analysis on the level of ease of the usage of this system.

Based on all the gathered data we can then draw conclusions about the system, related to customer satisfaction level or in the easiness of usage of the system, among other findings.

For this step to provide us relevant information it is crucial that our evaluation method comprises a series of questions that allow us to obtain this data.

Regarding the evaluation with users it contains the execution of a series of tasks covering all system features.

Doing this will allow to gather the opinion of the sample users about these features. Based on these data we can then make changes, if necessary or possible.

There was also the idea of performing a heuristic evaluation of the usability of the interface[5] but this was not possible due to time constraints.

4.1. User Interface Evaluation

Evaluation with users has the objective of understanding if, system users can use it an efficient, fast and concise manner.

The main objective of the system is that it is properly implemented to be as intuitive as possible for users, to facilitate their day-to-day work.

To make this assessment it was drawn a set of tasks to be performed by users, their performance was recorded based on various metrics.

Finally, the results were compared with the results of a user "expert".

Please note that all users who perform the tests had some previous experience with previous versions of the system.

After the execution of the tasks each user was asked to answer a questionnaire designated as "System Usability Scale" (SUS)[1].

This questionnaire consists of 10 questions, they are all phrases where the user indicates in a scale of 1 to 5, the level of agreement with the expressed phrase, where 1 means that you strongly disagree and 5 means that you are in full accordance with the statement.

Each question has an assigned rating and SUS results may vary between 0 and 100.

4.2. Results

As the users proceeded to the execution of the tasks the registration of four components, relevant to the assessment of each one of these was performed. It was also defined the expected results arising from the execution of the tasks mentioned above, by an "expert" user.

To conduct a deeper analysis, the two most important parameters were chosen:

- Time
- Number of clicks

To proceed to the analysis of time spent by each user on the execution of tasks, the following graph is presented.

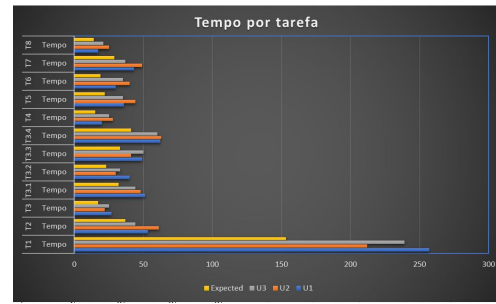


Figure 3- Time by task

We can observe that the user closest to the expected results, was the user number 3 approaching more of the results obtained by the "expert".

Regarding the number of clicks the analysis, gave the following chart.



Figure 4- Clicks by task

It appears from the figure above that the user 3 is again the user who was the closest to the expected results.

The remaining users obtained very positive results, the observed differences may be due to:

- Fatigue, because the test was performed at the end of a working day.
- Number of errors.
- Attention failure.
- Difference in the level of familiarization with the system.

The following phase concerns the results obtained by evaluating the system's usability.

The results were calculated for each user according to the assigned ratings.

The following figure presents the results obtained.

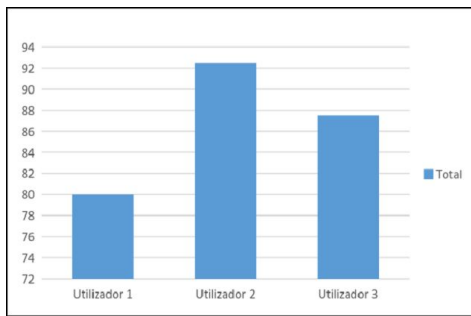


Figure 5- SUS Score

According to the figure above it can be derived that the values are equal or greater than 80 and the average thereof is approximately 86.67.

According to industry standards values below 51, translate into a rapid need to prioritize the improvement of system usability.

Values close to 68 mean that the system is within the desirable usability scale but there is room for improvement. Scores above 80.3 show that the users like the developed system and the likelihood of recommending it to others is quite high.

5. CONCLUSION

During this project technologies such as JavaScript, MongoDB, Angle, Express and the Google API were analysed.

Three solutions were presented to solve the problem analysed and the comparison between them was performed explaining why the final choice.

The adopted solution aims to reduce stress accumulated by users in carrying out their tasks, creating an alternative to the system used by the association which based on the usage of Excel files.

The development of this system also serves as proof that it is possible to build an information system utilizing Google Drive as a database.

Throughout this project plenty of challenges were found for the use of Google Drive as a database since it was not designed for this purpose, but these challenges were overcome, and the system developed has the following features, divided into three modules:

- Families' Management
- Animal Management
- Volunteer Management

Unfortunately, there was not enough time to implement every desirable function, so the following topics were left undone:

- Reporting (E.g.: Number of animals at risk playback)
- Allow the introduction of images to illustrate in the preview of animals.
- Information alerts (E.g.: Animals in need of vaccines in near future)

- Inventory Management Implementation.
- Improved functionality of creating a Technical Support Evaluation record.

Finally, it is recommended to those who are considering the usage of the Google API, to inform themselves first of its limitations, then to check the current version and the type of support available for it.

For the development of this project, it was necessary to use version 3 of Google Sheets instead of version 4 as the latter version has very little documentation until the time present, so the amount of support available was very limited.

REFERENCES

1. John Brooke. Sus-a quick and dirty usability scale. Usability evaluation in industry, 189(194):4-7, 1996.
2. Jeff Sutherland Darrell K. Rigby and Hirotaka Takeuchi. Embracing agile. Harvard Business Review, 2016.
3. Asim Liaquat. Google sheets v3 library. <https://github.com/asimlqt/php-google-spreadsheet-client>, November 2016.
4. Peter Membrey, David Hows, and Eelco Plugge. MongoDB Basics. 2014.
5. Jakob Nielsen. Usability inspection methods. In Conference companion on Human factors in computing systems, pages 413-414. ACM, John Wiley Sons, 1994.
6. Mahadev Satyanarayanan. Pervasive computing: Vision and challenges. Personal Communications, IEEE, 8(4):10-17, 2001.
7. Mark Weiser. Some computer science issues in ubiquitous computing. Communications of the ACM, 36(7):75-84, 1993.