

HEURISTIC EVALUATION RESULTS

Table of contents

Objectives.....	2
Heuristics principles used.....	3
Prioritizing problems.....	4
Results.....	5
1. Visibility of the system status.....	5
2. Match between system and the real world	6
3. User control and freedom	7
4. Consistency and standards.....	7
5. Help users recognize, diagnose and recover form errors.....	8
6. Error prevention	8
7. Recognition rather than recall	8
8. Flexibility and efficiency of use	8
9. Aesthetic and minimalist design.....	8
10. Help and documentation	8
11. Skills.....	8
12. Pleasurable and respectful interaction with the user	9
13. Privacy	9
Order of fixing.....	10

Objectives

The purpose of this analysis is to discover possible errors of the application related to the user interface that might hinder its use, spoiling the experience and overall satisfaction of the users.

The main goal, shared with the others usability tests performed, is to achieve an optimal implementation and usable application, easy to use and satisfying all the needs and expectations of users.

Heuristics principles used

To perform the heuristic evaluation, the principles defined by Jakob Nielsen have been used:

The 13 heuristics employed are:

1. Visibility of the system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose and recover from errors
10. Help and documentation
11. Skills
12. Pleasurable and respectful interaction with the user
13. Privacy

Prioritizing problems

As explained in the heuristic evaluation section of the memory, problems encountered should be classified according: the severity of the problem:

0. I don't agree that this is a usability problem at all.
1. Cosmetic problem only: need not be fixed unless extra time is available on project.
2. Minor usability problem: fixing this should be given low priority.
3. Major usability problem: important to fix, so should be given high priority.
4. Usability catastrophe: imperative to fix this before product can be released

Results

Once completed the heuristic evaluation, the errors that have been detected using the heuristic principles, are sorted depending on the severity.

1. Visibility of the system status

#	Problem	Severity	Heuristic	Solution
1	Icon design and style is not completely unified.	1	1.2	In the coded app there will be a major overhaul to unify even more all the icons.
2	If a task i.e. update the data values takes longer than normal, it should notify the user that it is still in process.	2	1.26	Add extra indicators if a task is taking longer than expected.

The problems detected do not affect usability, however they reduce the user experience and this is not a satisfactory outcome.

The first problem, it is not an actual problem, as is a planned to be fixed while coding the application, by using a unified library of icons.

The second problem is not very serious, however it requires some extra checks on the code to adapt it; to be able to detect if there is some delay and display the animation, to notify the users.

2. Match between system and the real world

#	Problem	Severity	Heuristic	Solution
1	Icons concrete and familiar	1	2.1	In the coded app there will be a major overhaul to unify even more all the icons.
2	Menu choices ordered in the most logical way.	2	2.2	Fix the menus, sorting alphabetically the options
3	Menu choices ordered in the natural sequence	2	2.3	Fix the menus, sorting alphabetically the options
4	Related and independent data showing on the same screen.	2	2.4	While coding the application, ensure a way to no show the data if not available, as it is not capture in a certain location.
5	Use of uncomment letter sequences.	3	2.18	In the prototype cardinal positions for wind are used, however for the coded application, it is going to be changed by icons.

Most of the problems detected do not affect usability, however there is one of them problem that greatly reduces the user experience, if the users do not know the abbreviations of the cardinal points.

The first problem, it is not an actual problem, as it is planned to be fixed while coding the application, by using a unified library of icons.

The second and third problems are not very important and they are easy to fix, just a proper sorting of the menus is required.

The fourth problem it's debatable, as a decision has to be taken, to decide if it's best to hide that bit of information with no values or show it as not available for that location.

The fifth problem would be the most serious, as it greatly affects the usability of the application if the users do not know the abbreviations for the wind direction, however this is a planned modification for the coded application.

3. User control and freedom

In this principle no problems have been detected affecting the usability or user experience.

4. Consistency and standards

#	Problem	Severity	Heuristic	Solution
1	Abbreviations do not include punctuation.	1	4.3	Modify all the abbreviations to finish with punctuation.
2	Use no more than four to seven colours	2	4.30	As of now many colours are used for the icons, but in the coded application all will be unified in 1 style
3	Abbreviations words with different lengths	2	4.43	As of now while using cardinal points for the wind direction, however in the applications this abbreviations will be replaced by icons.

The problems detected do not affect usability, however they reduce the user experience and this is not a satisfactory outcome.

The first problem, it is not a serious problem and it is easy to fix, just needs to fix all the abbreviations.

The second problem, it is not an actual problem, as is a planned to be fixed while coding the application, by using a unified library of icons, with unified colours.

The third problem is not an actual problem, as the standard abbreviations for the cardinal point are being use, however this is a planned modification for the coded application.

5. Help users recognize, diagnose and recover form errors

In this principle no problems have been detected affecting the usability or user experience.

6. Error prevention

In this principle no problems have been detected affecting the usability or user experience.

7. Recognition rather than recall

In this principle no problems have been detected affecting the usability or user experience.

8. Flexibility and efficiency of use

In this principle no problems have been detected affecting the usability or user experience.

9. Aesthetic and minimalist design

In this principle no problems have been detected affecting the usability or user experience.

10. Help and documentation

This section, cannot be properly evaluated as the actual help interface has not been developed.

11. Skills

In this principle no problems have been detected affecting the usability or user experience.

12. Pleasurable and respectful interaction with the user

#	Problem	Severity	Heuristic	Solution
1	Each icon is part of a family of icons	1	12.1	In the coded app there will be a major icon overhaul to unify even more all the icons.
2	Excessive details in icons	2	12.2	The coded application will have new unified icons, more simple and stylized.
3	Too much colour	2	12.3	After replacing the icons much less variety of colours will be used

The problems detected do not affect usability, however they reduce the user experience and this is not a satisfactory outcome.

All these problems are not actual problems, as they are planned to be fixed while coding the application, by using a unified library of simple and more stylized icons, with unified colours.

13. Privacy

In this principle no problems have been detected affecting the usability or user experience.

Order of fixing

After analysing the errors, it can be ensured that the user experience is not greatly affected, despite not being perfectly satisfactory, however all these errors, need to be fixed in order to make the application more usable, while providing a great user experience.

Below is the order in which they have to be solved:

Problem	Severity	Heuristic	Solution
Use of uncomment letter sequences.	3	2.18	In the prototype cardinal positions for wind are used, however for the coded application, it is going to be changed by icons.
If a task i.e. update the data values takes longer than normal, it should notify the user that it is still in process.	2	1.26	Add extra indicators if a task is taking longer than expected.
Menu choices ordered in the most logical way.	2	2.2	Fix the menus, sorting alphabetically the options
Menu choices ordered in the natural sequence	2	2.3	Fix the menus, sorting alphabetically the options
Related and independent data showing on the same screen.	2	2.4	While coding the application, ensure a way to not show the data if not available, as it is not captured in a certain location.
Abbreviations words with different lengths	2	4.43	As of now while using cardinal points for the wind direction, however in the applications this abbreviations will be replaced by icons.
Use no more than four to seven colours	2	4.30	As of now many colours are used for the icons, but in the coded application all will be unified in 1 style.
Too much colour	2	12.3	After replacing the icons much less variety of colours will be used

Excessive details in icons	2	12.2	The coded application will have new unified icons, more simple and stylized.
Icon design and style is not completely unified.	1	1.2	In the coded app there will be a major overhaul to unify even more all the icons.
Icons concrete and familiar	1	2.1	In the coded app there will be a major overhaul to unify even more all the icons.
Each icon is part of a family of icons	1	12.1	In the coded app there will be a major icon overhaul to unify even more all the icons.
Abbreviations do not include punctuation.	1	4.3	Modify all the abbreviations to finish with punctuation.