

University Degree in Computer Science and Engineering  
2018-2019

*Bachelor Thesis*

“Voice-based Mobile Application for  
Indoor Navigation”

---

Fernando García Sanz

Andrea Bellucci

Javier Triviño

Leganés, 2019



This work is licensed under Creative Commons **Attribution – Non Commercial – Non Derivatives**



## SUMMARY

This thesis covers the development of GuideMe, a mobile web application developed during an internship process in IBM.

The application allows locating many of the rooms which belong to IBM facilities in Madrid. This is done by means of either written or spoken natural language sentences, making use of artificial intelligence services. These services are provided by IBM cognitive proprietary software IBM *Watson*.

Firstly, an approach to the problem is done, as well as to the proposed solution. Previous work done related to the issue and the development and improvement of the different technologies employed to solve it has been studied and is also discussed along the thesis.

Throughout the document, different sections which cover the whole development of the project will be described: the first approach to the solution, the scheduling and work techniques and methodologies used by the development team, the different technologies employed during the project completion, the coding stage and the final solution and future lines of work.

This memoir also includes an economic analysis of the resources employed in GuideMe completion and the solution impact in social and economic environments, as well as a system requirements evaluation for both functional and non-functional ones.

**Keywords:** Speech processing, Indoor navigation, Web services, Chatbot, IBM Watson



## DEDICATION

To my parents, Fernando and Carmen, who have always trust me during the whole university stage, since the day I decided what I wanted to study and where, until today.

To my brother, Pablo, who has shown me the real meaning of strength, effort and overcoming. You have been a reference for me and your example has lead me during my career.

To my tutor, Andrea Bellucci, for accepting my thesis proposal and helping me with all the issues which came across its development.

To IBM, to all my coworkers, for giving me the opportunity of doing this project together with my university and helping me to reach my goals.

To all my friends of *La Vendicion Devs*, who motivated me during all these years and without any doubt, will keep doing it in the future.



# CONTENTS

1. INTRODUCTION. . . . .	1
1.1. Motivation of Work . . . . .	1
1.2. Problem Formulation . . . . .	1
1.3. Solution Justification. . . . .	2
1.4. Regulatory Framework . . . . .	2
2. STATE-OF-THE-ART . . . . .	4
2.1. Location Services . . . . .	4
2.2. Machine Learning . . . . .	6
2.3. User Interfaces . . . . .	7
2.3.1. Accessible Interfaces. . . . .	9
3. DESIGN OF SOLUTIONS. . . . .	13
4. DEVELOPMENT OF THE SOLUTION . . . . .	15
4.1. <i>Agile</i> Methodology. . . . .	16
4.2. <i>Scrum</i> Framework . . . . .	16
4.3. Design of the Application . . . . .	18
4.4. Application Coding . . . . .	22
4.4.1. IBM <i>Watson</i> . . . . .	22
4.4.2. Maps Development . . . . .	25
4.5. Application Final Implementation . . . . .	25
4.5.1. Application Testing and Security Scan. . . . .	34
4.5.2. Application Release at IBM. . . . .	37
5. PLANNING AND BUDGETING . . . . .	39
5.1. Cost of the Project . . . . .	39

5.2. Work Planning . . . . .	40
5.3. Socio-Economic Environment . . . . .	42
6. CONCLUSIONS . . . . .	43
6.1. Objectives Met . . . . .	43
6.2. Future Lines of Work . . . . .	44





## LIST OF FIGURES

2.1	<i>Google Maps</i>	4
2.2	IBM <i>Deep Blue</i> by Jim Gardner is licensed under CC-BY	7
3.1	GuideMe <i>Design Thinking</i>	13
3.2	GuideMe empathy map	14
4.1	GuideMe <i>GitHub</i> panel	15
4.2	<i>Mock up 1</i>	18
4.3	<i>Mock up 2</i>	19
4.4	GuideMe initial flowchart	20
4.5	GuideMe final flowchart	21
4.6	IBM <i>Watson</i> services implementation	24
4.7	GuideMe main page	26
4.8	GuideMe non-valid room	27
4.9	GuideMe valid room	27
4.10	GuideMe first step	28
4.11	GuideMe second step	29
4.12	GuideMe <i>NPS</i> site	31
4.13	GuideMe use case	32
4.14	GuideMe testing stand	35
4.15	IBM security scan	36
4.16	GuideMe interview cover	37
4.17	GuideMe final stand	38

5.1	GuideMe iterations' analysis . . . . .	41
5.2	GuideMe Gantt chart . . . . .	41
6.1	<i>NPS</i> outcome . . . . .	43
6.2	Functional requirements consistency matrix . . . . .	57



## LIST OF TABLES

6.1	GUIDEME-FR-FUNC-001 . . . . .	46
6.2	GUIDEME-FR-FUNC-002 . . . . .	47
6.3	GUIDEME-FR-FUNC-003 . . . . .	47
6.4	GUIDEME-FR-FUNC-004 . . . . .	48
6.5	GUIDEME-FR-INST-001 . . . . .	48
6.6	GUIDEME-FR-INST-002 . . . . .	49
6.7	GUIDEME-FR-INP-001 . . . . .	49
6.8	GUIDEME-FR-INP-002 . . . . .	50
6.9	GUIDEME-FR-INP-003 . . . . .	50
6.10	GUIDEME-FR-INP-004 . . . . .	51
6.11	GUIDEME-FR-INP-005 . . . . .	51
6.12	GUIDEME-FR-NPS-001 . . . . .	52
6.13	GUIDEME-FR-NPS-002 . . . . .	52
6.14	GUIDEME-FR-NPS-003 . . . . .	53
6.15	GUIDEME-FR-WAT-001 . . . . .	53
6.16	GUIDEME-FR-WAT-002 . . . . .	54
6.17	GUIDEME-FR-WAT-003 . . . . .	54
6.18	GUIDEME-FR-WAT-004 . . . . .	55
6.19	GUIDEME-FR-WAT-005 . . . . .	55
6.20	GUIDEME-FR-WAT-006 . . . . .	56
6.21	GUIDEME-FR-WAT-007 . . . . .	56
6.22	GUIDEME-NF-INST-001 . . . . .	58

6.23	GUIDEME-NF-INST-002	58
6.24	GUIDEME-NF-INST-003	59
6.25	GUIDEME-NF-INST-004	59
6.26	GUIDEME-NF-INST-005	60
6.27	GUIDEME-NF-INST-006	60
6.28	GUIDEME-NF-DESIGN-001	61
6.29	GUIDEME-NF-NPS-001	61
6.30	GUIDEME-NF-NPS-002	62
6.31	GUIDEME-NF-GDPR-001	62
6.32	GUIDEME-NF-SEC-001	63



## GLOSARY

UC3M	Universidad Carlos III de Madrid
IBM	International Business Machines
GuideMe	IBM application for guiding users inside Madrid facilities
GitHub	Cloud code storage system with a version control system
NPS	Net Promoter Score
GNU	GNU's Not Unix. Collection of free software and Unix type OS
HTML	Hyper Text Markup Language
CSS	Cascade Style Sheets
Sass	Syntactically Awesome Style Sheets
Design Thinking	Meetings to develop a solution to an isolated problem
Brainstorming	Open-ended proposal of ideas
Iteration Showcase	Meeting to decide next iteration workload and review requirements
CIO	Chief Information Officer
Inkscape	Vector graphics open-source software
DB2	IBM proprietary database language
PL/SQL	Procedural Language / Structured Query Language
Gantt Chart	Bar chart which illustrates a project schedule
LOPD	Organic Law of Data Protection (Ley Orgánica de Protección de Datos)
GDPR	General Data Protection Regulation
Watson	IBM artificial intelligence and cognitive services proprietary software
WAI	Web Accessibility Initiative





# 1. INTRODUCTION

The world is changing quite fast, as well as people way of living. Nowadays, cities are eventually bigger, as are their buildings. People move one place to another very often due to several reasons, such as work, family... They do not remain settled in one place as they used to do before.

Due to the increasing number of people traveling, it has been necessary to improve location services, since people usually move to places they have never been before, so getting lost is a quite common issue. It might seem that this is not an important problem, but the amount of time and resources people employ trying to get back on the right track is enormous if we study it from a global point of view.

## 1.1. Motivation of Work

Getting lost is something common when a person goes to a place she has never been before. If the structure of the place is messy or if its size is big enough, finding a specific place might be an arduous task, even more if we have no details about it or if its description or the path one is not adequate. The human being tends to wander around trying to find the exact place, and if the search is not successful, he will end up asking someone who knows the place how to get to the goal venue.

## 1.2. Problem Formulation

IBM main facilities in Madrid are located in *Santa Hortensia 26* street, consisting of a huge building with numerous stories in which is pretty difficult locate yourself if it is the first time you visit the venue. Being the main operations center in Spain, it is usually a quite crowded building, to which both clients and vendors of IBM come from all over the world. With many people working for IBM and those looking forward making business with them, but who are not familiar with the building, locating a room can be arduous.

IBM building has a reception just in the main entrance of the building, but the person-

nel who works there is usually busy due to many other tasks, so it is difficult to attend the volume of people accessing the building and be focused on other labors.

### **1.3. Solution Justification**

In the month of September 2018 I was selected by IBM, one of the biggest computing businesses of all time, to be part of the CIO team as IT specialist intern. My role consisted of developing technological solutions which could ease IBM workers tasks within the enterprise. At that moment, the team I am part of was commissioned to develop a system which could facilitate finding a specific room within IBM facilities.

IBM CIO department decided to design an application to address this issue. By means of this application, the user should be capable of finding the desired room, avoiding creating queues in front of the reception desktop, speeding up the whole process and allowing reception staff to focus on other tasks.

The proposed solution should comprise the biggest amount of people possible, so accessibility has been taken into account, since it should be possible to communicate with the application either typing or talking to the device. It should be capable of interpreting the user input, such as a normal person does, and provide a correct path to reach the goal room. In this way, it is no needed the user to learn specific commands, the application will be in charge of translating the input into useful data.

As said, the solution has to be be accessible, intuitive and easy to use, avoiding overwhelming the reception staff and providing a faster solution, since possibly the staff do not know the location of each single room in the building.

### **1.4. Regulatory Framework**

As the application records data for processing the room requested by the user, it could compromise security if any other person is capable of extracting or visualizing the data introduced by another user.

To solve this problem and to be compliant to the *Organic Law of Data Protection* [1], the application will automatically delete the input data once the user finalizes the

complete interaction or each two minutes without interaction, when the page will refresh and delete all information stored in the browser.

A similar treatment happens with the ratings and suggestions left by a user: the data are totally anonymous and it is impossible to relate them to a particular user. The application will also refresh each two minutes without any kind of interaction, deleting the input information in case it was not submitted by means of the rating site.

Furthermore, the images directly related with IBM confidential data and facilities used in this document have been pixelated in order to prevent any data leakage which could compromise IBM privacy and security, as accorded in the original collaboration agreement with the enterprise.

According to the employed software, programming languages are under *GNU* licence and can be freely used. *IBM Watson* services and the deployment platform *IBM Cloud* are under IBM software licence.

## 2. STATE-OF-THE-ART

In order to provide a solution to the raised problem, several technology fields and techniques are required. The main concepts which take part in the process are the followings.

### 2.1. Location Services

Traditional maps were a valid solution during a wide time period, but they have several important drawbacks: when you get a paper map, you are getting a representation at the time the map was printed, and it will never change, it will never be updated according to changes in cities arrangement, new road routes, changes in streets names, etc.

New technologies have made traditional maps become obsolete. They deal with the main drawbacks a printed map has quite effectively. These services are connected to the internet, so their developers can continuously update their content, so they can be much more reliable, true to reality. Many quality products have appeared in the market, all of them trying to get their piece of cake. Some of them are world-known, such as *Google Maps*<sup>1</sup> or *Apple Maps*<sup>2</sup>, which are multipurpose applications.

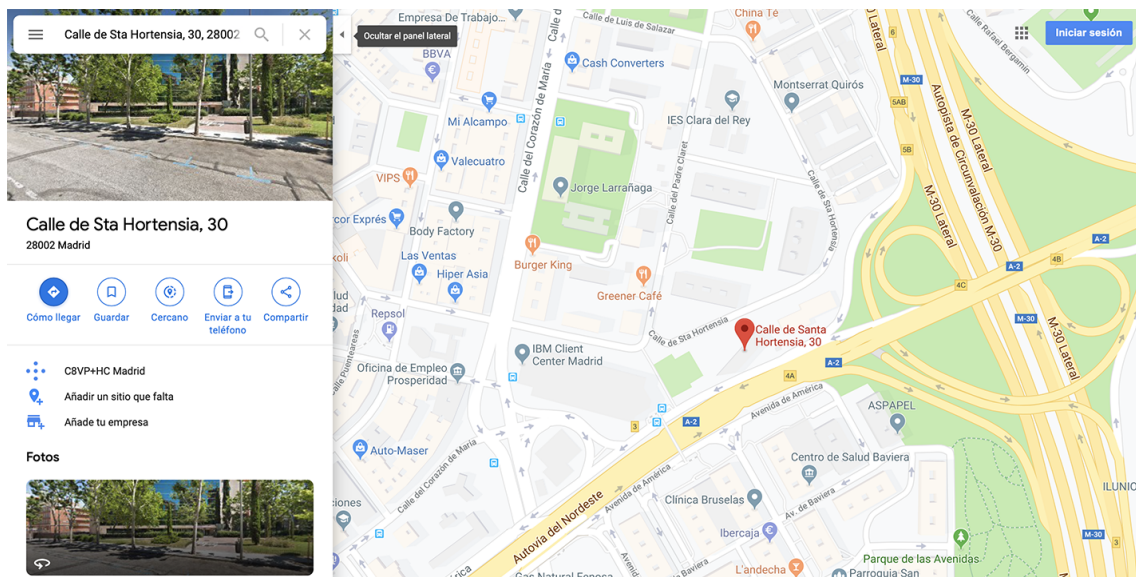


Fig. 2.1. *Google Maps*

<sup>1</sup><https://maps.google.com/>

<sup>2</sup><https://www.apple.com/es/ios/maps/>

Others, such as *Waze*<sup>3</sup> or *Here Maps*<sup>4</sup> are mainly focused on vehicle displacements, generally providing a much deeper information than the first named do.

As it can be seen, navigation and location technology has evolved a lot, but it is mainly focused on outdoor environments. Currently, some buildings can be perceived as mazes, specially when people do not know them, and there is a technological gap in location services for indoor navigation, since buildings' layouts are needed to develop these solutions and private companies do not usually share them to external businesses, due to confidentiality reasons.

Moreover, traditional *GPS* systems are not available, or their performance is highly reduced, in indoor environments, due to the existing difficulty to the signal in order to trespass solid objects.

As mentioned, the main difficulties related to indoor location are *GPS* malfunction in these environments and a pre-deployment of systems which requires exhaustive maps, which highly increase the cost of these solutions.

These issues have promoted research over this field. In "*Indoor Location Without The Pain*" [2], in order to avoid the pre-deployment cost of developing the necessary maps, *WiFi* networks, cell phones and *GPS* technology are used. Users will move around the facilities in normal course and the *WiFi* signal strength received in their mobile phones will be recorded and sent to a central server which will process it and perform a mapping of the indoor environment.

Other researchers take advantage of the solid materials which belong to indoor environments to develop indoor location systems, as it can be found in "*Indoor Location Sensing Using Geo-Magnetism*" [3]. The researchers performed a location measurement by means of the disturbances caused by the structural steel elements of a building. The obtained results were quite satisfactory, but as the previous solution, the needed materials to deploy it make the solution cost increasing a lot.

Outdoor location services, such as those named above, have eventually evolved, integrating new functions such as traffic prediction, public transport and private enterprises integration as a transporting way. It is logical to think that this trend will not change,

---

<sup>3</sup><https://www.waze.com/es/>

<sup>4</sup><https://wego.here.com/?lang=es-es/>

making these services much more powerful and complete. The next step to take consists of keep improving indoor location systems in order to cheapen them and develop reliable enough solutions.

## 2.2. Machine Learning

Nowadays, machine learning is one most popular topics, as well as one of the most promising fields of technology, being integrated in many different fields, and providing satisfactory results in all of them. Machine learning has enhanced quite a lot the possibilities current services offer.

Machine learning consists of studying different kinds of statistical models, patterns and data in order to extract information which allows systems to perform a task without explicit instructions. This can have multitude of applications in many diverse fields, from the own field of computer science to other ones such as medicine, engineering, mathematics, etc.

Its beginning as a concept began around the year 1950, when mathematical and statistical methods were refined and started to be applied for this purpose. *Bayes' Theorem* and *Markov Chains* are the main basis of this research. Alan Turing proposed in 1950 a kind of machine capable of learning, and in 1951 the first neural network machine was built by Marvin Minsky and Dean Edmonds. Shortly after, IBM started its research in machine learning.

Around the 60s, *Bayesian Methods* for probabilistic inference started to be applied in machine learning and new algorithms were discovered, but machine learning research progress was not as good as expected, which led to several years of stagnation. Nevertheless, several discoveries aimed to retake the research in machine learning field.

In 1997, a huge inflection point is found, when *Deep Blue* [4], a machine learning system built by IBM, beat Garri Kasparov, chess world champion, showing the power of machine learning against a human being [5]. At the same time, research in neural networks and algorithms had progressed a lot, making machine learning a quite powerful tool.



Fig. 2.2. IBM *Deep Blue* by Jim Gardner is licensed under CC-BY

During the beginning of the 2000s, eventually more companies started to be interested on machine learning. Recently created businesses such as *Google* and *Facebook* became powerful machine learning research pillars. During the 2010 decade, a new concept, called deep learning arrived, which allowed to apply machine learning to many different systems and software solutions.

### 2.3. User Interfaces

All of these above mentioned technological improvements will be nothing without a proper interface which eases user interaction with the machines and applications. A good user interface (*UI*) will enhance the user experience (*UX*), fundamental factor in the usability of a component. If the *UX* is good, the user will prefer that application among others which offer the same functionality, but whose usability is worse.

User interfaces have vary a lot since their origin. In the beginning of computing, computers were programmed, so all existing interfaces were programming interfaces [6]. First computers users were programmers who used them to either program instructions, perform calculations or start developing in the programming languages that exist at that



time.

It was not until 1963 when Douglas Engelbart introduced the concept of the "computer mouse" [7]. Engelbart presented this concept during the "Mother of All Demos" in 1968. Douglas Engelbart proposed many ideas related to computing too, some of them of great importance, being precursor of Moore's Law [8] five years before.

A few years later, a new concept was introduced. Touchscreens are a wide spread concept nowadays, but their invention was patented in 1969 by E.A. Johnson, introducing the idea in "*Touch Display — A Novel Input/Output Device for Computers*" [7].

But, after all of these inventions, a true inflection point according to user interfaces is found in the beginning of the 70s. The first graphical user interface (*GUI*) was developed in the Xerox Palo Alto Research Center (Xerox PARC). The *Alto* was released in 1973 and implemented the mouse concept explained before, revolutionizing the concept of user interfaces. This was Steve Jobs' inspiration to release the *Apple Lisa* and eventually, the Macintosh computers.

The Macintosh system included revolutionary ideas such as presentation software (PowerPoint basis). These computers were intended to the main public, as personal computers, concept which was widely exploited during those years by businesses such as IBM.

During the following years, a huge development according to user interfaces was experienced. It is also fundamental to talk about Don Norman, who established the rules for developing a good user experience, based on a user centered design [9].

Within these last two decades, computer systems have evolved a lot. Numerous improvements have been introduced and new concepts have been designed, such as modern operating systems, smartphones, modern laptops and all-in-one computers, and user interfaces have evolved a lot as well, being one of the main improvements the usage and recognition of voice commands.

During the whole process of development of user interfaces, voice implementation has been eventually improved.

The first implementations happened during the 50s and 60s decades, when Bell Labs and IBM built systems capable of recognising numbers from 0 to 9 and 16 different En-

glish words respectively [10].

Next years, due to the implementation of the *Hidden Markov Model* and US Department of Defense research, the quality and amount of recognised words by these systems was widely enhanced.

A great example about voice implementation in interfaces can be found in "*Put-That-There*" [11]. It consisted of a system which implemented both voice-input and gesture-recognition on a large raster-scan display. Voice and gestures are combined in order to perform commands, taking into account pronouns and variables to reference items shown in the display.

The system was implemented inside a room of the size of a personal office. The device's vocabulary can recognize up to 120 different words, size which could be extended up to 1000 words.

By means of commands such as "*Create..., move..., etc*" and pointing to the position with "*here or there*", the user was able to trigger changes in the interface. The system will process the voice commands and the place pointed by the user, performing the desired operation. The user was able to even perform operations relating different elements, such as "*Move the blue triangle to the right of the green square*".

This system was a huge revolution, demonstrating the possibilities of voice commands and recognition.

The years before entering the new century there already were systems capable of continuous speaking recognition. During the following years, current famous systems were released, such as *Siri*, *Apple* voice recognition system, presented in 2007, which was later integrated on *iPhone* devices in 2011, *Google Voice Search* and, more recently, *Amazon Echo* and *Google Home*, where it can be observed the huge improvement of voice user interfaces, wide spread in day by day life.

### **2.3.1. Accessible Interfaces**

For an interface to be effective, it has to be properly used by the maximum amount of people possible, and to success in this goal, the concept of accessibility has to be taken into account.

The concept of "reaching the maximum amount of people possible with a system design" is what is known as *Universal Design* or *Design For All* [12].

This idea was first thought as developing an environment obstacle-free, so the greatest possible amount of people could enjoy it. This idea was later developed in the environment of digital interfaces, looking forward applying those principles to user interfaces development [13], inasmuch as a program which could reach a bigger amount of users will be beneficial to both developers and clients, since more people will use it and, therefore, developers will get more users.

### **Voice Recognition Systems**

Voice recognition has a crucial role in interfaces' accessibility. Due to the improvements done in voice recognition and in user interface techniques, many systems can be used by a wider amount of people, since its interface allows people with certain disabilities to interact with them properly.

One of the main improvements has been the text-to-speech and its opposite speech-to-text technologies. An exhaustive study of these technologies can be found in "*Review of text-to-speech conversion for English*" [14]. It can be seen that the first hardware systems related to this goal appeared around the 50s, evolving until getting the first commercial systems in the decade of the 80s.

Current systems perform an analysis of sentences, splitting them into words and these into phonemes, which can be interpreted as sounds, later concatenated by different protocols, compounding the sound equivalent of the original written sentence.

On the other way around, speech waveform is analyzed by speech-to-text systems, getting the different audio patterns and obtaining its equivalent graphic symbol. Those symbols are concatenated, compounding the equivalent text to the input speech.

Nowadays, those systems use different approaches, algorithms and technologies, such the improvements above mentioned in *Machine Learning* to enhance the possibilities and accuracy of these solutions.

There are several solutions which implement these technologies to enhance systems' accessibility.

In "*JustSpeak: Enabling Universal Voice Control on Android*" [15] a system is implemented in *Android* devices, allowing system interaction via voice commands, making the device fully accessible to blind people. The system uses speech recognition and parses the results into commands. Then, it looks for actionable elements in the device screen related to the commands and manipulates the one which matches the most.

Another solution developed by *Google* and which is currently implemented in *Android* devices is *Talkback*<sup>5</sup>. In "*Getting Smartphones to Talkback: Understanding the Smartphone Adoption Process of Blind Users*" [16], it can be seen how this technology has been implemented in smartphones, with several blind participants involved in the study process. Their feedback has been useful to find barriers that could not be found in task-based settings, to study *Talkback* learning curve and to learn how a better integration of support tools can be done in smartphones.

Nowadays, *Talkback* also implements gestures in *Android* devices and can also be used in *Google Chrome* web browser.

*Apple* has also implemented text-to-speech systems in their smartphones, being capable of dictating text in thirty languages<sup>6</sup>.

These are only some examples of how speech recognition can enhance systems accessibility, being a field which will have a wide participation in future years.

## **Voice Assistants**

Artificial intelligence improvements as well as voice recognition ones have made possible voice assistants development.

Voice assistants are a hot topic right now. *Google Assistant*<sup>7</sup> and *Alexa*<sup>8</sup> services, respectively developed by *Google* and *Amazon* are widely used systems nowadays by millions of people [17].

These systems utility goes from making a phone call to the remote control of connected devices, easing interaction and enhancing the possibilities and actions a user can

---

<sup>5</sup><https://support.google.com/accessibility/android/answer/6283677?hl=en>

<sup>6</sup><https://developer.apple.com/accessibility/ios/>

<sup>7</sup><https://assistant.google.com>

<sup>8</sup><https://developer.amazon.com/alexa>

do by means of voice commands.

The developed solution discussed in this thesis uses IBM *Watson*, taking advantage of machine learning services to improve and perform a more accurate voice recognition, which will be used to properly achieve the application objective.

### 3. DESIGN OF SOLUTIONS

In this stage, once the problem has arisen, IBM CIO team decided to look for the most proper solution to the issue.

To do so, both the team in charge of developing the solution and IBM people external to that team, arranged a *Design Thinking* [18] meeting. The purpose of this kind of meetings is to isolate the main problem, reinterpret it, giving it the most adequate definition and propose solutions to the issue.

During the *Design Thinking* a process of *Brainstorming* is performed. *Brainstorming* [19] consists of all members of a group provide ideas with the purpose of solving a question or problem. The main point of this technique is looking for the biggest amount of ideas possible. It goes for quantity and not for quality, because it has been proved that in most of the cases, among low quality ideas, a brilliant one may appear. Once all ideas have been expressed, they are classified according to feasibility, cost and time factors. Similar ideas can be combined in order to achieve a more complete concept.



Fig. 3.1. GuideMe *Design Thinking*

In IBM CIO *Design Thinking* meetings, *Brainstorming* is totally open-ended, not directed in any sense, so anyone can provide the number of ideas he considers and the content of this ideas is free too, no words or concepts have the obligation of been chosen.

To provide a more specific view about the problem, the team also developed an empathy map, which is a kind of diagram where one possible user of the solution is represented, specifying his background. Thanks to it, it is possible to know more deeply which could be the main points in which the solution has to focus to satisfy the user worries.

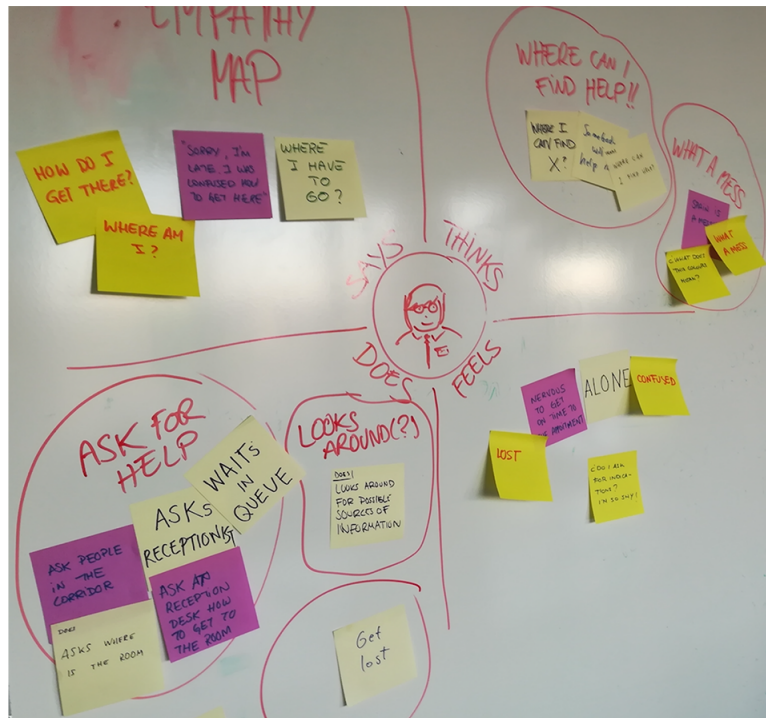


Fig. 3.2. GuideMe empathy map

Once all ideas are expressed, they are combined and classified in the way described before.

Firstly, expensive ideas that do not deal properly with the arisen problem are discarded. Those which deal properly with the problem and have a small cost are the first of been taken into account. Among these ideas, they are also classified according the required time for their completion and their feasibility. Then, these ideas are ordered according to their importance in the global compute of the solution. Those of huge importance need to be completed as soon as possible, meanwhile those of lower importance are allowed to be delayed in benefit of the priority ones completion.

Once all this process has finished, the team is capable of extract the main objectives the solution must satisfy as soon as possible, those which can be delayed but has to be completed and the ones which have to be discarded due to their solution focus.

## 4. DEVELOPMENT OF THE SOLUTION

After the content explained in the previous section, it was decided that the company needed an application which could guide any kind of user with the minimum number of steps possible from the entrance of the building to any area of the facilities. The application to be developed must be a web application so IBM services could be easily implementable and it could also be more scalable among different devices if necessary. It was at that time when the name of the application, GuideMe, was decided too.

To start the solution development, all requirements were included in a specific project created inside *GitHub*<sup>9</sup> platform, where a backlog column containing all the requirements extracted from the *Design Thinking* meeting was created. In that backlog column, all requirements are ordered by priority and, as said before, those with the highest priority are the ones intended of being finished before.

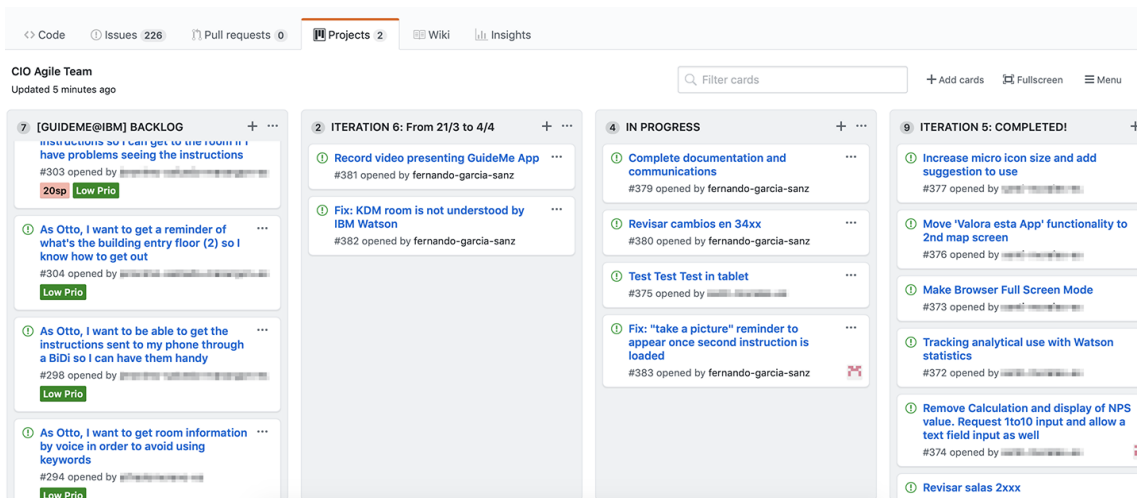


Fig. 4.1. GuideMe *GitHub* panel

A wider scope of all requirements specified the *Design Thinking* can be found in *Annex: Requirements*.

Furthermore, to maintain an active communication during the project development, *Slack*<sup>10</sup>, a corporate communication tool, was used, employing the professional plan pro-

<sup>9</sup><https://github.com>

<sup>10</sup><https://slack.com/intl/es-es/>



vided by IBM.

To understand the way of working in IBM which is applied to the development of this project, it is necessary to talk about *agile* methodology and *scrum* framework.

#### **4.1. Agile Methodology**

*Agile* is a working methodology based in an iterative and incremental approach by means of sprints. It is devised for projects whose requirements and needs eventually evolve, in an unpredictable way [20].

This working methodology is particularly devised for software development, due to the continuous changes of requirements, either functional or non-functional, software projects usually suffer. This allows a continuous support from the development team to the product and as well as a persistent feedback from the stakeholders to adapt the product to their needs in every moment.

*Agile* was devised to overcome the usual issues software projects experimented years ago: the stakeholders present their needs and worries, the development team (or person in charge) accepts the offer once a deadline is agreed and the development team starts working on the product. When the deadline is near, the team presents the product to the stakeholders, who have a lot of complains, requests, etc. This is mainly due because in large software projects, as well as in non-software projects, since the project is agreed until the deadline is reached, needs, interests and requirements may vary a lot, and the requested final product is far away from the initial agreed one.

Here is when *agile* appears. It allows, as said, a continuous feedback in both ways, from the development team to the stakeholders and vice versa.

*Agile* can be applied in many different ways, but maybe the most common and effective is through the *scrum* framework.

#### **4.2. Scrum Framework**

*Scrum* is defined as:

*“A framework within which people can address complex adaptive problems, while*

*productively and creatively delivering products of the highest possible value” [21].*

*Scrum* delivers efficiency within a product development. *Scrum* implements a series of rules which bind together roles, events and artifacts.

Everyone involved in the framework receives a role:

- Product owner: in charge of maximizing product’s value
- Development team: group of people in charge of delivering a potentially releasable increment of the final product at the end of each sprint
- *Scrum* master: responsible of promoting *scrum* within the product development

The workload is organised in sprints, slices of time, of two weeks in this case, in which the potentially releasable increment of the final product is crafted. Sprints are consecutive, starting the next just in the end of the previous one. Sprint objectives has to be set at the beginning and no changes are allowed which may endanger the set goals. Within each sprint, a daily *scrum* meeting of short duration is arranged in order to plan the workload of the whole day and inspect the development of the sprint goals.

At the end of each sprint, a sprint review is held, in which the achieved is analyzed, adapted to the backlog if necessary (changes in the backlog might had happen). During this meeting, both *scrum* team and stakeholders collaborate about what has been done during the sprint to adapt the outcome to the objectives as much as possible.

Finally, there are three concepts which need to be taken into account:

- Product backlog: Ordered list of everything which is truly needed in the product
- Sprint backlog: Set of the product backlog to be done during the sprint plus a plan of how to do it to reach the sprint goal.
- Done set: It defines when the work is done for a product increment. The increment is usable, so the product owner may be capable of releasing it if desired.

Now that it is known what *agile* and *scrum* mean and their differences, it is possible to continue explaining the different phases of the solution development.

### 4.3. Design of the Application

According to the different requirements extracted from the previous meetings, the development team was in charge of developing a first *mock up* of the application. This *mock up* will be the basis of the application design, providing a first view of the design ideas to be later modified if necessary.

The application was devised to be used in an *iPad*, so the design must be optimized for this platform and dimensions. For that reason, the *mock up* was developed based on *iPad* dimensions and applications design.

Therefore, the first iteration consisted of developing the first *mock up* of the application, which can be seen below:

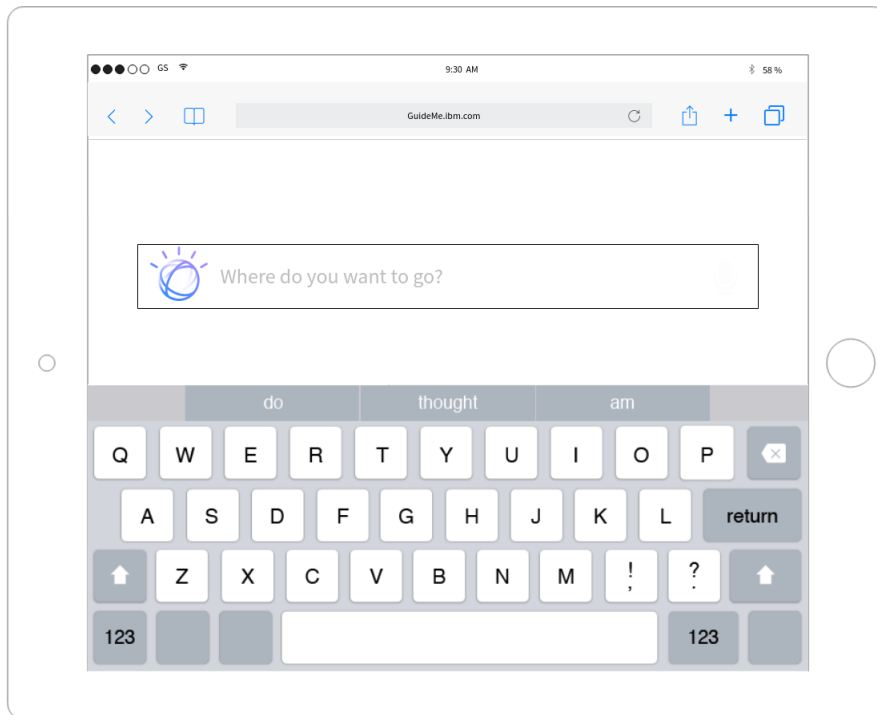


Fig. 4.2. *Mock up 1*

This represents the main page of the application, where the user has to introduce the details of the room he is looking for. Then, by means of IBM *Watson* artificial intelligence services, the application will be capable of understand the user input and provide him a set of instructions about how to reach the target room.

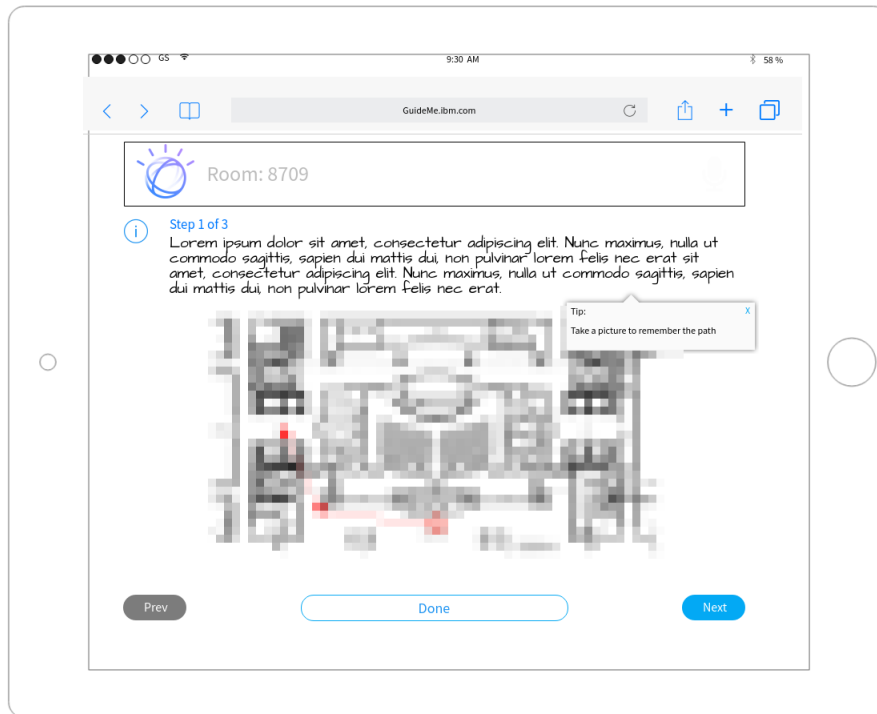


Fig. 4.3. *Mock up 2*

This second image represents the instructions provided by the application to get to the input room. Several things can be seen here:

- A panel at the top which indicates the input room, so the user can know if the displayed route takes him to the room he wants.
- A set of instructions divided into different steps (3 in this case), to make easier comprehend the provided directions.
- A huge image where the route to follow from the stand in which the *iPad* is located is shown.
- A tip on the top right corner of the picture remembering the user to take a picture of the instructions to remind them.
- A set of buttons which allows the user to advance into the next instruction, return to the previous one and finish the execution if he already knows how to get to the room or simply wants to leave the stand.

Once the *mock ups* had been designed, during the meeting at the end of the iteration

they were presented to the stakeholders, who approved the idea, so the application *front-end*, the visual and design part, can start to be coded.

The first application functional idea can be seen in this flowchart:

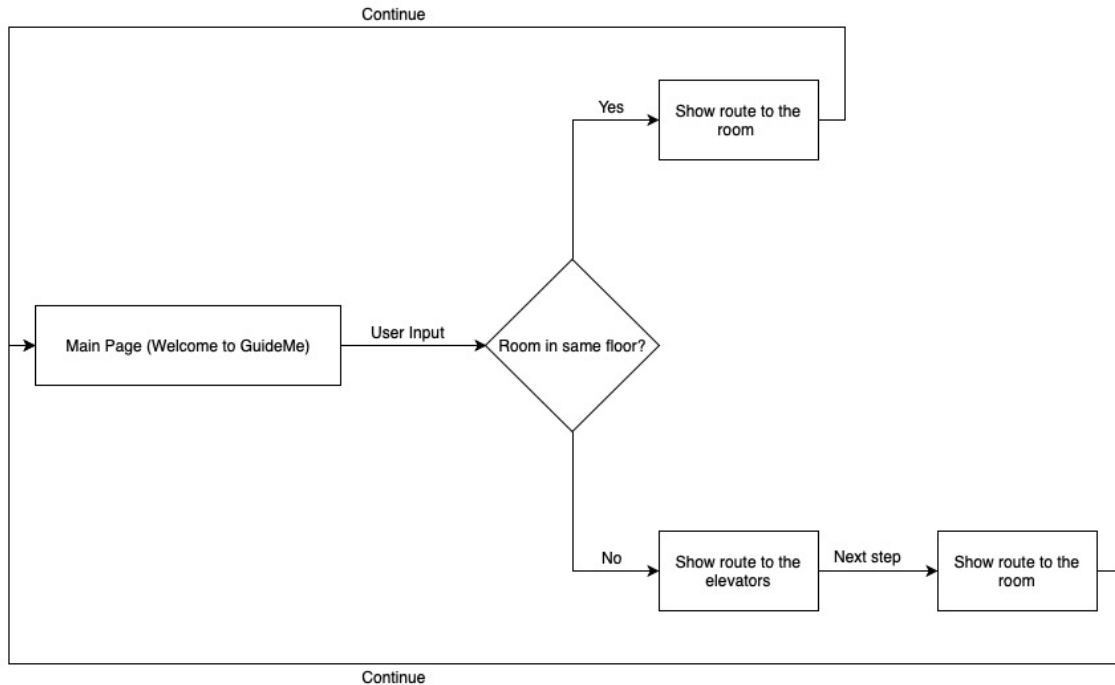


Fig. 4.4. GuideMe initial flowchart

Eventually, during the next iterations, already agreed visual elements were coded and added to the application so the user can use them during the iteration. Moreover, during the following meetings, new ideas and requirements were added to the application by the stakeholders, so the final functional implementation of the application, which can be seen in the following flowchart, is far away from the first design:

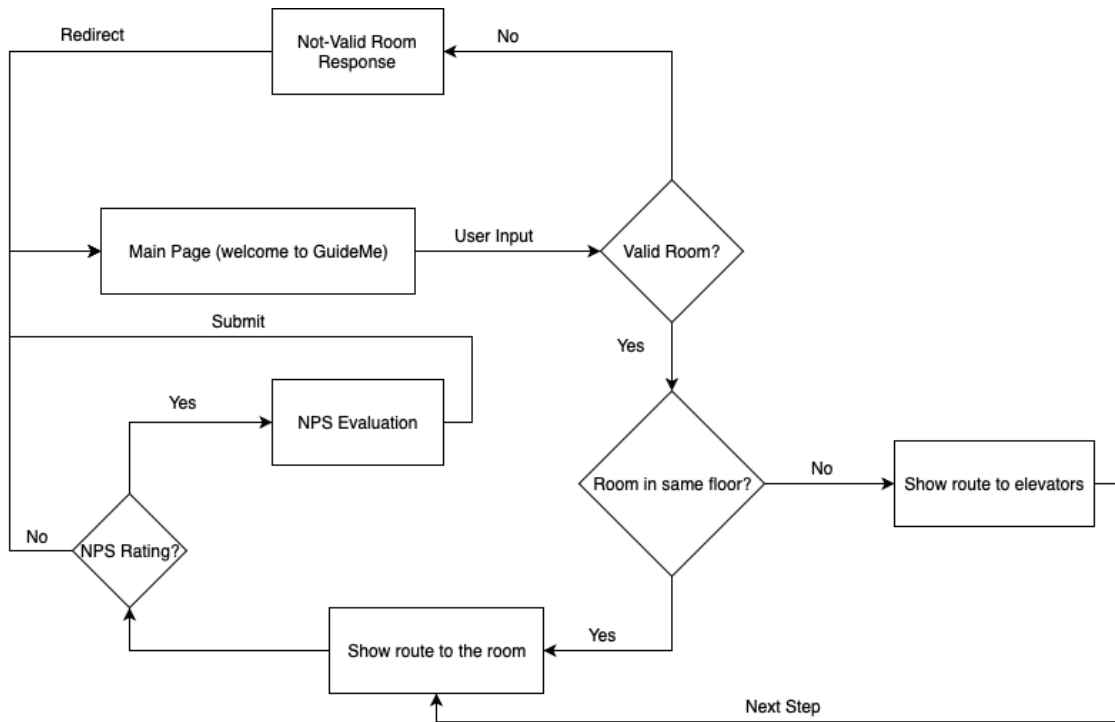


Fig. 4.5. GuideMe final flowchart

As it can be seen, the flow through the application is quite simple. It all starts in the main page, where the user introduces the data referencing the target room. Then, the application will analyze the input and decide whether the input room is a valid one or not. If it is not, it will response with a non-valid room response, not loading any instructions set, as the room is not between the valid ones. On the other hand, if the input was a valid room, GuideMe will load the instructions site and decide whether to show the instructions in one or two steps, according to the floor in which the target room is located. If it is in the same storey the stand is, then just one instruction page will be shown, telling explicitly how to get to that room from the stand. If it is not, then the application will tell the user how to get to the closest elevator and to which floor he needs to go, and then, in the next step, how to get to the target room from the elevators' exit at that storey.

Finally, when the complete set of instructions is traversed, a rating option appears in the last page. If the user decides to rate the application, he will be redirected to a new page in which he can submit a rating from 0 to 10, which will be evaluated using the *NPS* rating mechanism, and also leave a comment or suggestion related to his experience with the application.

Whether deciding to rate the application or not, once the user presses over the *continue*

button, the application will load the main page, starting a new iteration.

According to the flowchart, three main web pages were developed. The first one corresponds to the main page, where the user introduces the input, the second one corresponds to the instructions page, developed in a dynamic way so there are not a huge number of different pages due to the different rooms in the facilities, but just one site which will adapt to the user input, displaying the proper instructions and images in each case, and the third one, in which the user will be able to rate the application if desired.

#### **4.4. Application Coding**

Once the full design and information flow inside the application have been decided, it is time of creating the real product.

For the application development, as it is a web application, the programming languages selected were *HTML*, *CSS* and *JavaScript*.

*HTML* is used for defining the content of each of the different activities the web application consists of, *CSS*, implemented by means of *Sass* language, defines the style each visual component of the application will have, and *JavaScript* is used to define the logical sentence which will provide functionality to the own application, making it useful to solve the arisen problem.

Moreover, the implemented IBM *Watson* services were programmed in *node.js*, which allows to execute *JavaScript* commands in a server.

In later iteration showcases during the application development stage, it was decided that the system also needs a database which records ratings and suggestions from users, using this feedback to provide a better service and support. The database was developed in *DB2*, an IBM proprietary database language, based on *PL/SQL*.

##### **4.4.1. IBM Watson**

The main technology used in this application is IBM *Watson*. It is the true inflection point in GuideMe development, because it is the entity which provides the "intelligence" it needs to properly understand users' inputs.

IBM *Watson* is an artificial intelligence technology developed by IBM. It is mainly focused, but not bounded, on natural language understanding, which means to understand properly inputs, orders and conversations provided in natural language, the one used by humans to understand each others. Therefore, IBM *Watson* allows communicating without knowing a series of specific commands, being much more accessible to a widest amount of users.

Due to IBM *Watson* characteristics, it can be defined as a cognitive system.

*“Just as humans do, cognitive systems are driven to understand things by decomposing expressions of an idea and then combining that with context.”* [22].

The continuous development of IBM *Watson* has allowed that it has not been only applied to conversations understanding, but a much wider amount of fields. It has been employed in medicine, mathematics and different scientific scenarios [23].

IBM *Watson* services can be implemented in on-development applications thanks to *node.js*, deploying the different needed packages in the application, provided by IBM *Watson* service online platform.

It is only needed to perform a connection to the service provider from those packages and modify the provided skeleton to adjust the code to the needed functionalities as much as possible. The basic *Watson* service can be enhanced by means of stacking different additional services which provide different functionalities.

This service works remotely, sending the information to a central computer in which information is processed and retrieving the result to the client device.



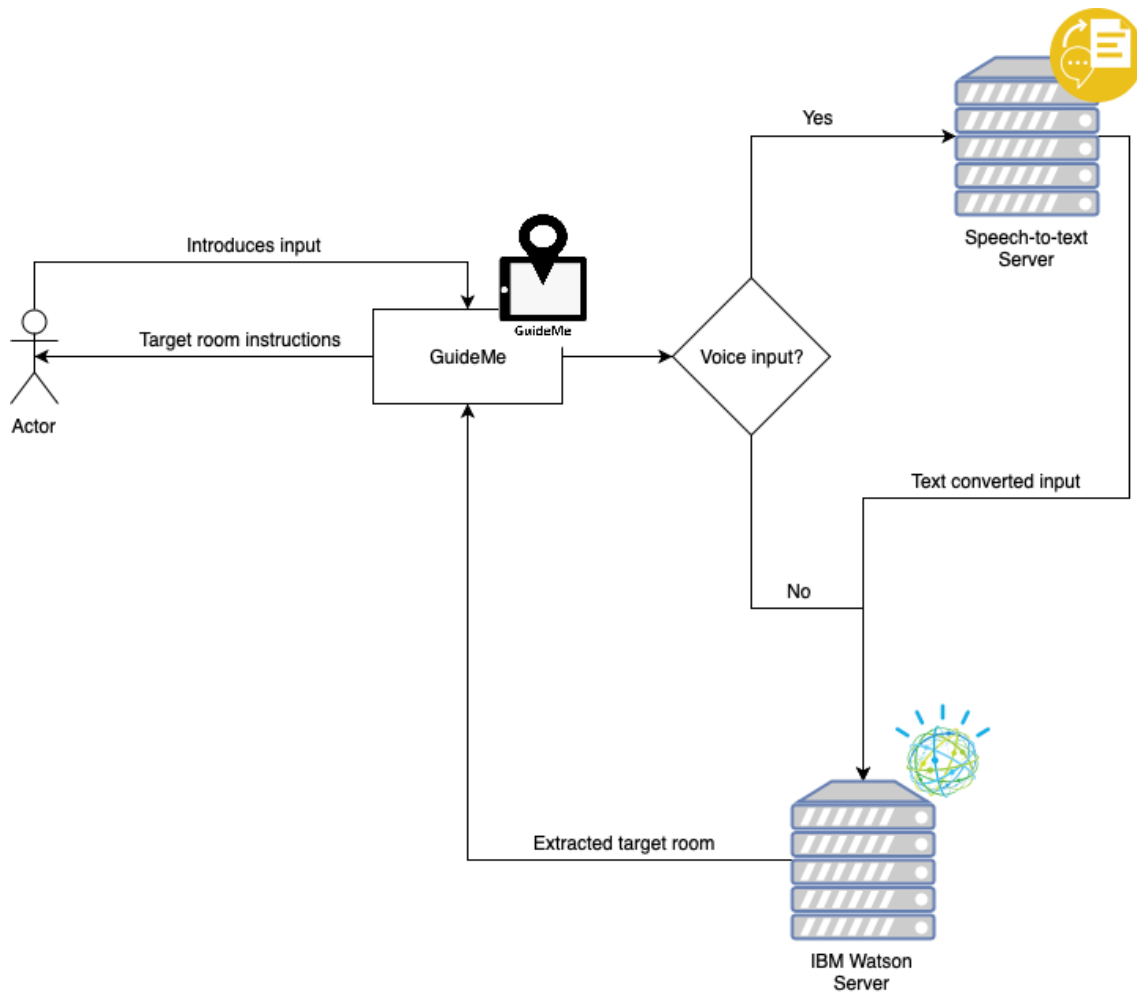


Fig. 4.6. IBM *Watson* services implementation

As it can be seen, the application integrates IBM *Watson* services the following way:

- The user introduces the input either via typing or via speaking through the microphone.
- If the input is via voice commands, it is sent to IBM *Watson* speech-to-text server by means of an implemented *API*, which will perform the connection and message passing. Speech-to-text server will translate the voice input to a written one and will send it back to GuideMe.
- The written input is sent to IBM *Watson* server, which will recognize the target room from the whole input and return it to GuideMe, or if it was not possible, an error message will be returned.

- GuideMe will receive the target room and will process the proper instruction set to get to it, or it will return an error response in case IBM *Watson* was not capable of getting it from the text input, asking the user for reintroducing the input.

#### **4.4.2. Maps Development**

Besides the different technologies and programming languages employed in GuideMe development, it was also necessary to draw the different maps of each IBM building floor.

To do so, first a reconnaissance tour was done by the development team, with the main goal of taking into account the disposition of the different rooms, the layout of each floor and the location of services and facilities which may be potentially included in the final implementation of GuideMe.

Once the principal features related to each floor were recorded, maps' drawing could start.

The main idea was that the drawn maps had to be reliable but simple, avoiding any distraction their design and content might provoke. Maps were crafted by means of *Inkscape*<sup>11</sup>, a open-source vector graphics software.

IBM building is composed by ten different floors, and each of them is usually split into nine different sections. All different routes were drawn once the final emplacement of GuideMe stand was known, drawing a specific map for each route, which leads to a section in a specific floor. Therefore, more than eighty maps were drawn to display all possible routes recorded in GuideMe.

#### **4.5. Application Final Implementation**

The first design of the application basically consisted of a main page, the one explained in Fig. 4.5, and the instructions one, which also appears in the previous figure. These two sites can be displayed either in English or Spanish, the two languages in which the application is developed, allowing the user to freely use GuideMe in the language he prefers.

---

<sup>11</sup><https://inkscape.org/es/>

The final design of the main page is the following one:

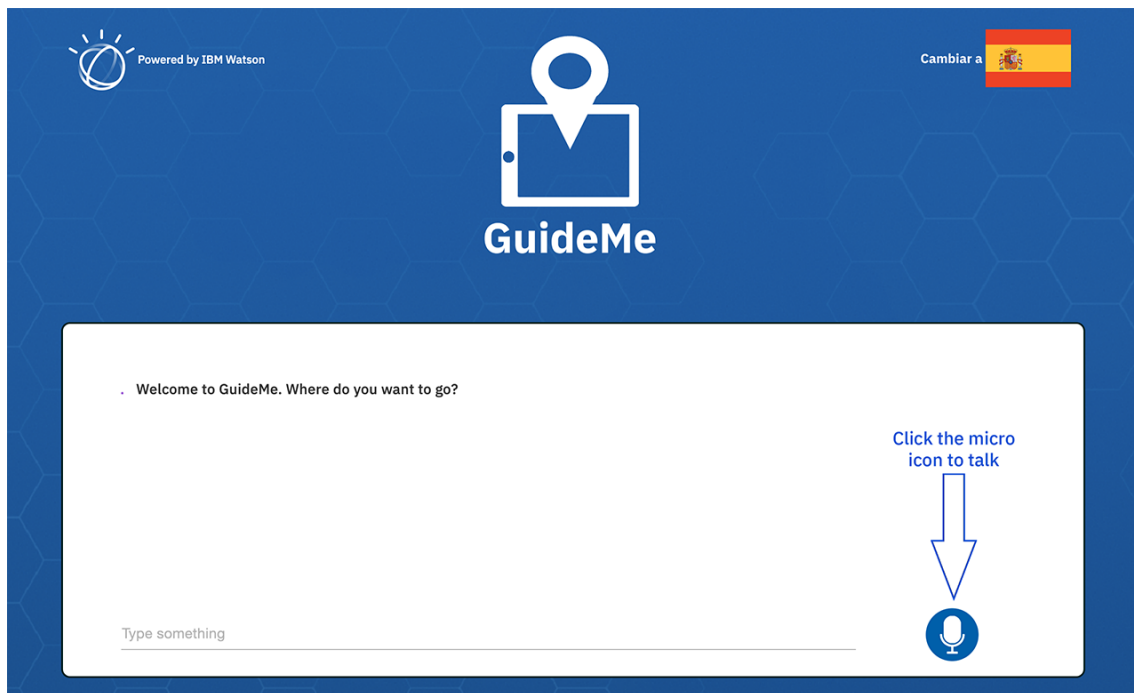


Fig. 4.7. GuideMe main page

As it can be seen, the main page includes several functional elements:

- **Flag icon:** This icon allows the user to swap between English and Spanish interface of the application just by clicking on it.
- **Main chat box:** A huge chat box is displayed in the main page, where the user has to type the input he wants to introduce to the application. There, it can also be seen what is understood by the speech-to-text service when introducing the information via voice input. The input will be shown in form of chat message when sent to *IBM Watson* platform and it will also reply in form of chat message with the answer to the requested room.
- **Microphone icon:** This icon activates the voice input in GuideMe. When the icon is pressed it will change its colour to green, pointing that it is recording the input, and when the service detects that the input has concluded, it will change back to blue and the message will be sent, being also shown in the chat box as when a typed input is sent.

Following the information flow in the application, once the user introduces an input, either by voice or typing, the input sentence is evaluated. If the target room is not a valid one, either because the user did not specify any room or because the room does not belong to the valid set, the outcome of the application will be the following:

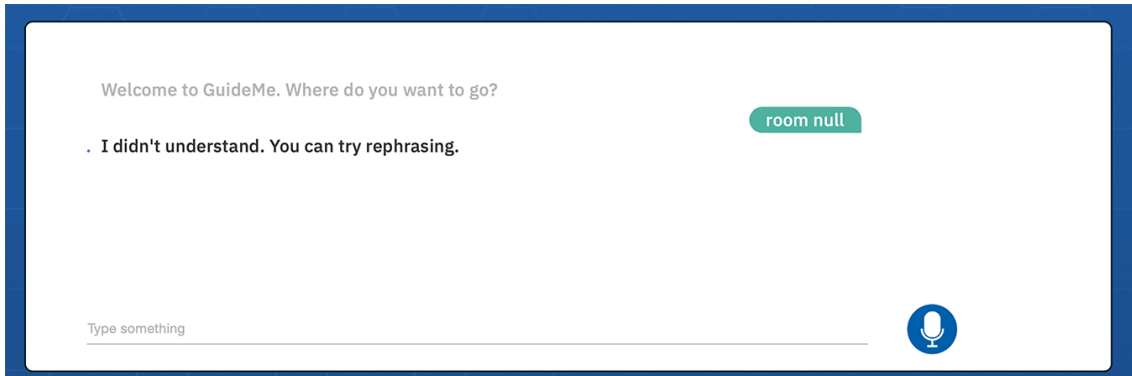


Fig. 4.8. GuideMe non-valid room

So the user will be warned about a non-valid input, asking him to re-introduce a valid one.

If the input is valid, the following will be shown:

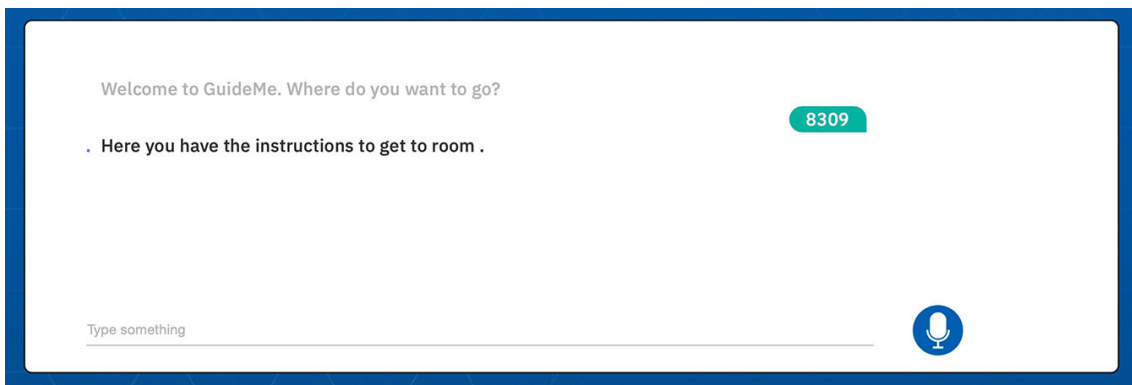


Fig. 4.9. GuideMe valid room

After introducing a valid input, a success message will be shown in the chat box and the user will be automatically redirected to a new site in which the instructions about how to get the target room are shown.

As explained before, depending on the floor the target room is located, instructions are displayed either in one or two steps. This is known by means of internal functions

of the application, which parse the target room and depending on the structure it follows, determine the floor the room is located.

The first step displayed to get to the goal venue appears as follows:



Fig. 4.10. GuideMe first step

This first instructions page also contains several elements which have to be taken into account:

- Input room information: The first element which can be seen is placed at the top of the web page. It consists of a header in which are shown both the target room and the number of steps in which the information about how to get to the target room is going to be displayed. This element might be useful to the user, so he can ensure that the instructions shown are the correspondent ones to the room he truly wants to find.
- Instructions map: A huge image is shown in the middle of the site. It contains the route from the point the stand is located to the room, in case there is only one step in the instructions, or to the nearest elevators in case there are two. It is the main element in the screen so it can be easily discovered and most kind of users can also properly understand the instructions. Its design has been simplified as possible to

ease its content comprehension.

- **Instructions description:** In addition to the map, instructions are also described under it. It can ease its understanding in case the map is not enough, and vice versa.
- **Home and Next Step buttons:** These two buttons appear in the bottom left and right corners respectively. White colour for the background and blue for the letters have been used in order to provide a good contrast, so they can be easily found in the screen. Furthermore, a flashing animation has been also added to both of them, enhancing its notoriety in the site. The first button allows the user to go back to the main page of the site, in case he has introduced an erroneous room, the room understood by the program is not the proper one, etc. The second button allows to go load the second instructions' step in case there are two steps or finishing the execution and loading the main page of the site if there is only one step, being its content *Continue* instead of *Next Step* in that case.

In case there is a second step in the instructions, it will be displayed in the following way:

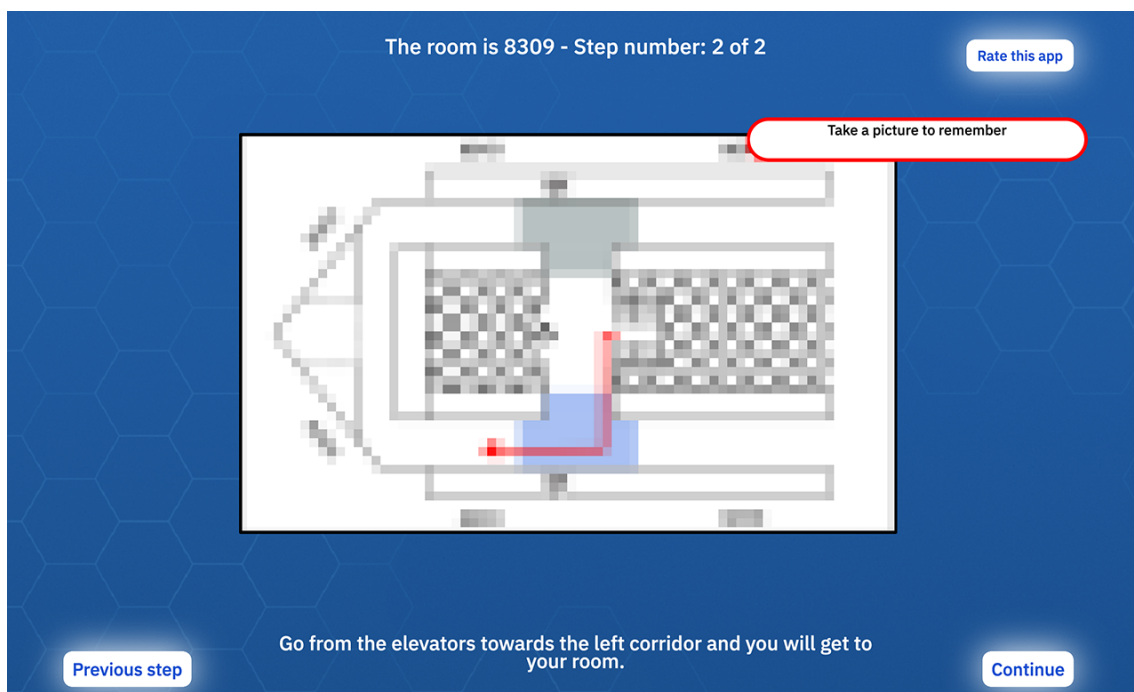


Fig. 4.11. GuideMe second step

This second step incorporates most of the components already explained in the previ-

ous figure Fig. 4.10, but some new elements appear here and others have changed:

- *Rate this App* button: Pressing this button will lead to the *NPS* rating site, which will be explained later on.
- Instructions map: In this case, the map will show the route from the closest elevators to the target room to the room itself in its correspondent floor. A colour code which represents each area is also shown in the image, helping the user to ensure he is on the right path to the venue.
- *Take a picture* sign: A warning which contains the text *Take a picture to remember* is displayed during five seconds, advising the user to take a picture of the content of the second step in the instructions, since it will correspond to the route from the elevators to the target room in the correspondent floor, and it is probable that the user forgets it during his way to the goal venue.
- *Previous Step* and *Continue* buttons: In this case, buttons content has changed. *Home* button has been substituted by the *Previous Step* one, which instead of loading the main page of the site, loads the first step of the instructions, restoring the previous content. *Continue* button substitutes the *Next Step* one and its function is the same explained before for the *Continue* button which appears in the one-step instructions set: loading the main page of the site, finishing the execution for the current user.

Now, the user can decide either rating the application or just finishing its usage. Pressing the *Continue* button will lead to the main page, which has already been explained before Fig. 4.7. Nevertheless, pressing the other button will load the *NPS* rating site, which looks as follows:

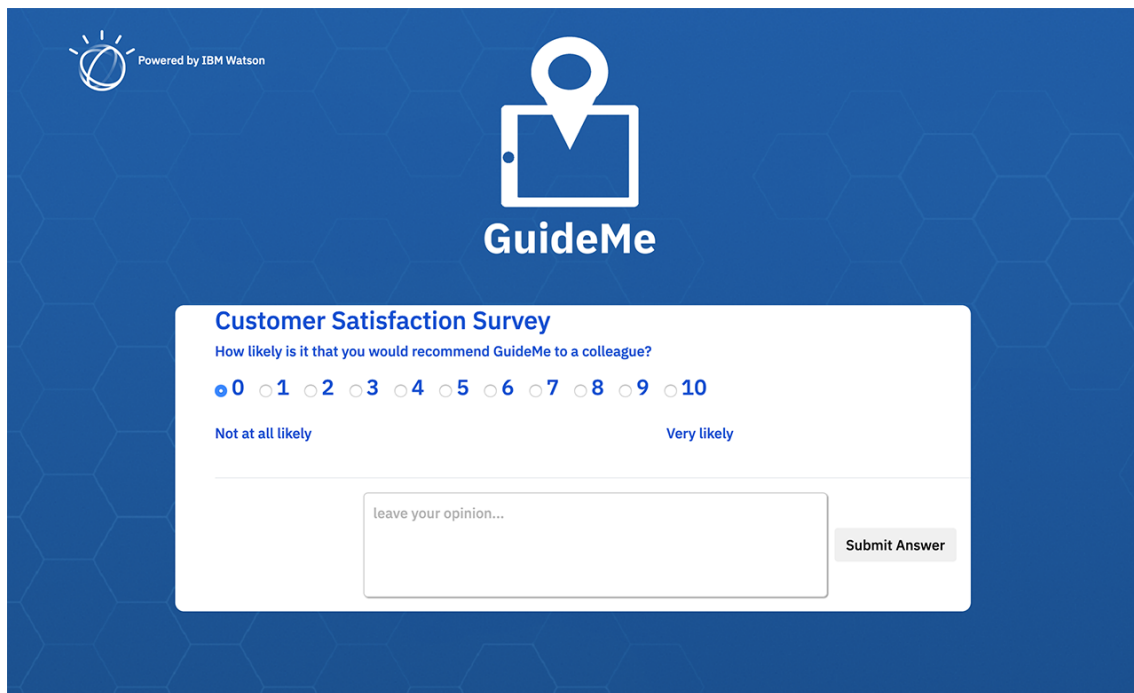


Fig. 4.12. GuideMe NPS site

This section is composed by a big box in the middle of the page, where the user can rate the application in a scale from 0 (the lowest rating) to 10 (the higher one). The user can also leave a comment of any kind, suggesting possible improvements, pointing out incorrect behaviours of the application, etc.

NPS scale works in the following way [24]:

- The user is requested a score between 0 and 10. Those who rate, the application in this case, with a score between 0 and 6 are considered *detractors*, which means that they were not happy when using the product, and his opinion might damage the product reputation as well as the brand one.
- Those who provide a score equal to 7 or 8 are considered *passive* users. They were satisfied by the product, but they are not enthusiastic about it, which means that they might be eventually attracted by similar solutions provided by competitors.
- Finally, those who provide a score equal to 9 or 10 are what is called *promoters*. They were not only satisfied by the product, but also are loyal product enthusiasts who will promote it among their contacts.



NPS final rating lays between -100 and 100, being -100 the equivalent rating extracted from a *detractor* score, 0 the one extracted from a *passive* user and 100 from a *promoter*. Therefore, a NPS rating above means that most of the users are not only happy with the product, but also will recommend it to their contacts.

Reaching this point, the full GuideMe information flow has been explained. A use case of the application can be seen in the following diagram:

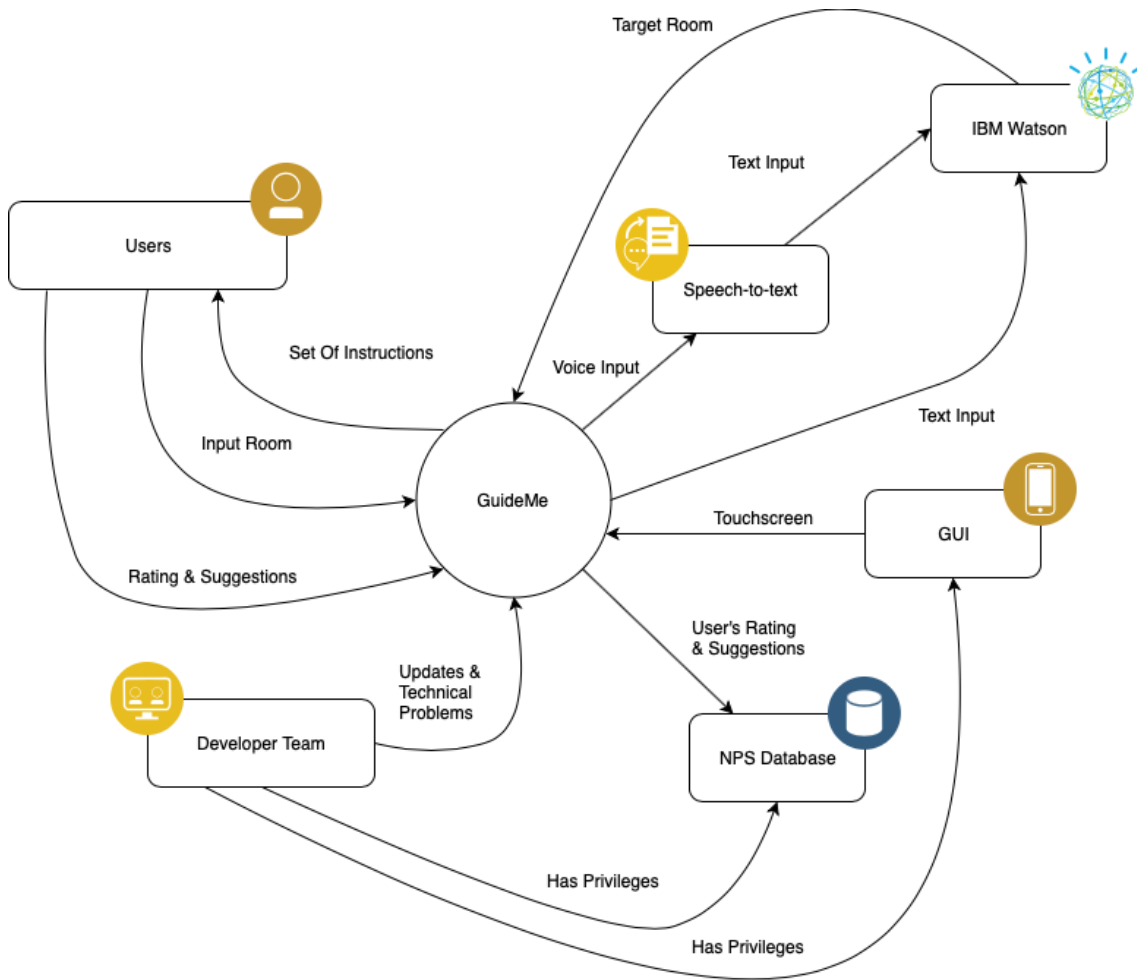


Fig. 4.13. GuideMe use case

### Problems During the Development Stage

The development stage of the application took most of the time employed in the project.

First of all, agreeing about the design of the application was not an easy task. Some of the stakeholders thought that the colours employed in elements such as buttons, the main page..., were not the proper ones, so different tests were made to agree about the

combination which fits the most to the biggest amount of people involved into the project development.

Implementing the different services offered by IBM was not an easy task. First of all, it was necessary to learn how to use each service properly, and this was done in controlled environments where they can be free tested.

IBM *Watson* implementation was not an easy task, since it has to be linked to the *API* servers which provide the service all time, and an incorrect connection will provoke a failure in the whole system, since it is the one in charge of translating user inputs to an understandable language for the application.

Adding the speech-to-text service, needed to translate voice inputs to written ones, to the IBM *Watson* one was not an easy task due to the need of locating different connections and services over the previous implemented system, increasing the possibilities of a service connection to fail in some moment.

The good news consist of as the speech-to-text is located over the IBM *Watson* interface, if the service fails, *Watson* service can still work properly with typed text.

Once both services were implemented and properly working, it was time to test it in the platform selected for its use inside the enterprise, and *iPad Air 2*.

Testing the application in the *Apple* devices it was proved that the only web browser which has permissions to access all the needed services from the device was *Safari*, due to *Apple* restrictions. Nevertheless, *Safari* has several drawbacks which made impossible to implement the solution as desired at the beginning in the *iPad*: the microphone did not work at the first attempt, it was impossible to load the web application in full screen mode, etc.

Several solutions were proposed at the moment. Some of them were tried, such as developing a web view application to load in full screen mode the web window, which made impossible using the microphone, or using different web browsers, with none of them providing permissions for the necessary *iPad* functionalities.

So, finally, it was decided to change the deployment platform from and *iPad Air 2* to a *Samsung Galaxy Tab A*, after checking that the main features of the application properly worked in an *Android* device.

Using a new device different from the one initially considered supposed a change in the design and proportions of the website, to adapt them as much as possible to the new tablet.

During this stage, some new elements were decided to be introduced by the stakeholders, such as a mechanism to record users' opinions and suggestions about the application using *NPS* rating. Moreover, different already included elements were changed or re-designed according to the new thoughts expressed by the project manager.

Finally, GuideMe application could be deployed in the *Android* device using a specific browser called *Fully Kiosk Browser & App Lockdown*<sup>12</sup>, which allows setting the browser in full screen mode and also accessing to the microphone, two of the main requirements of the application.

#### **4.5.1. Application Testing and Security Scan**

When the main development stage of GuideMe was almost complete and thanks to *agile* methodology, which ensures that at the end of each iteration, a functional, although not fully complete, solution is provided, GuideMe functionalities could be tested.

To do so, the application was installed in a development environment inside IBM cloud system. Just by means of the correct *URL* and a connection to IBM private network, a device could load the application website.

First, as mentioned above, the application was deployed over an *iPad Air 2*, and later on, the deployment device swapped to a *Samsung Galaxy Tab A*, due to non-affordable drawbacks found when testing over the first device. It was during this stage that the different issues mentioned in *Problems During the Development Stage* were found.

The tablet device was placed inside a stand composed by a base, a pipe traversed by the charging cable and a protective cover inside which the device was located.

---

<sup>12</sup><https://www.ozerov.de/fully-kiosk-browser/>



Fig. 4.14. GuideMe testing stand

It was freely tested by the people allowed to enter the room in which the stand was located, who communicated the different issues found during their interaction with the application, as they could be:

- Not locating a included in the system room.
- The second step in the instruction set to a room is not displayed.
- When looking for a room, the application displays the route to a different one.
- Voice commands are not properly recognized by GuideMe.
- Microphone is not working.
- *IBM Watson* does not reply to the input.

- Loading times are too high.
- Application does not load after internet connection is lost and later recovered.
- Application is not found in current *URL*.
- Spanish and English languages do not work properly.

Testers feedback was received either during iteration meetings or by mail or *Slack*. Some of the issues could be due to changes in the application version deployed in the development environment, which needs to be reloaded in each device it is used. Others could be caused by IBM *Watson* service failures, provoking both text and speech-to-text recognition not to work. The remaining ones could be provoked by code mistakes, bugs which needed to be solved as soon as possible, so the application could be deployed in the final production environment.

Furthermore, previous to the final deployment in the production environment and the application release, it was also necessary to perform a security scan over the already deployed in development environment application, ensuring that it contains the lowest possible number of vulnerabilities, and that those contained, if any, do not compromise the security of the system.



# IBM Application Security on Cloud Scan Security Report

Fig. 4.15. IBM security scan

Vulnerabilities are classified by the analysis tool as high, medium and low risk vulnerabilities. For an application to be deployed, it needs not to have any high risk vulnerability and that it successfully passes the security scan.

GuideMe did not contain any kind of high risk vulnerability, so it could be approved for deployment. Once the application was tested enough and considered fulfilled to deal with the main purpose it was thought, GuideMe was released at IBM.

#### 4.5.2. Application Release at IBM

Once GuideMe has been tested enough and it fulfills the requirements stated by the stakeholders, it is allowed to be released at IBM.

The release announcement was done by means of a communication to all IBM workers in the building via email, but previous to it, the communication team made an interview to the development team. The outcome of this interview can be found in the internal IBM network, and any worker is able to leave its comments and suggestions. On it, there are explanations about the need of this application, how it was developed and future plans about GuideMe, according to its technical support and a future enhancement of its possibilities.



Fig. 4.16. GuideMe interview cover

Once the stand was placed in its final location, the application was newly tested, finding out that *WiFi* connection did not have enough power to properly get to the device, due to the structural elements surrounding it (walls and columns). Therefore, it was necessary to move the closest network repeater from its original place to another one, closer to the stand emplacement, so it could properly connect to *IBM Watson* services, providing the expected behaviour.

A huge flag was placed next to the stand location. It was designed by the communication team and its function is to notify in a visual way where *GuideMe* stand is located, so it can be easily recognised by any potential user.



Fig. 4.17. GuideMe final stand

## 5. PLANNING AND BUDGETING

### 5.1. Cost of the Project

To accomplish the development of GuideMe, several resources have been necessary, as well as a development team. In this section, an analysis of all relevant items and entities who took part in GuideMe development is going to be performed.

First, according to the development stage of the application, it began with the *Design Thinking* meeting, held on 25<sup>th</sup> September, 2018, and the development was concluded during the month of May of 2019, so it took a total of 9 months if September and May are completely included in this interval.

During this period, the development team is monthly paid. The team is composed by 3 developers and the monthly salary of each of them is equal to €400. Therefore, the cost in salaries is:

$$€400 \cdot 3devs \cdot 9months = €10.800$$

Moreover, the development team needed computers to accomplish GuideMe coding and testing, as well as tablets for its testing and deployment.

The computers used were a *Lenovo T440*, a *Lenovo T480* and an *Apple Macbook Pro 2017*, with an approximated price of €570, €1.120 and €1.500. Therefore, laptops final cost is:

$$€570 + €1.120 + €1.500 = €3.190$$

Two *Apple iPad Air 2* with *WiFi* and *LTE* connection were used, as well as two *Samsung Galaxy Tab A*. Their respective approximated cost is €420 and €290. Therefore, tablets final cost is:

$$€420 \cdot 2 + €290 \cdot 2 = €1.420$$



Next, according to software, most licences were free, since IBM departments pay yearly for their own proprietary services, such as IBM *Watson*. Therefore, it was only needed to pay for the kiosk mode software used in the *Samsung* tablets. The cost was €5 per tablet:

$$€5 \cdot 2 = €10$$

Finally, the remaining costs correspond to the employed materials, such as the tablet stand and the advertisement flag located next to it. The stand cost is estimated around €70, meanwhile, the flag one is estimated around €90. Therefore, the total cost of these elements is:

$$€70 + €90 = €160$$

So, the total cost of the project development is equal to:

$$€10.800 + €3.190 + €1.420 + €10 + €160 = €15.580$$

## 5.2. Work Planning

As mentioned in previous sections, GuideMe was entirely developed during nine months, starting in September 2018 and finishing in May 2019.

It was also mentioned in sections *Agile Methodology* and *Scrum Framework* that *Agile* methodology and *Scrum* framework were used to develop the project scheduling. Therefore, as explained, the scheduling conformed by different iterations of the same length.

Nevertheless, according to the eventual development of the project, some iterations had to be modified due to bank holidays, the need of finishing a specific task within an iteration, etc.

With the project completion, it is possible to analyze the tasks performed in each iteration, specifying its content and the time employed in each of them. The outcomes of the analysis can be seen in the following table:

Iterations			
Start Date	End Date	Description	Duration (days)
25/9/18	9/10/18	Design thinking and mock-up development	14
10/10/18	24/10/18	First interface design implemented	14
25/10/18	8/11/18	Adjustment of interface design	14
9/11/18	23/11/18	Building reconnaissance and maps drawing	14
24/11/18	8/12/18	Learn Watson assistant and design improvements	14
9/12/18	7/1/19	Test and adjust everything on iPad	29
8/1/19	25/1/19	Improve maps. Multilingual Watson. First security scan	17
26/1/19	6/2/19	Link speech-to-text and external banner design	11
7/2/19	21/2/19	Review texts and maps. Adjust design to tablet	14
22/2/19	8/3/19	Fix microphone functionality. Update security scan	14
9/3/19	23/3/19	Implement NPS rating site. Browser full screen mode	14
24/3/19	7/4/19	Desing improvements and adjustment to new tablet	14
8/4/19	7/5/19	Last changes. Complete documentation. Final security scan	29
8/5/19	15/5/19	Application deployment. Test in final emplacement	7

Fig. 5.1. GuideMe iterations' analysis

These data have been used to develop a Gantt chart, a kind of bar chart which illustrates a project schedule.

In this chart, a more visual explanation of the content and the time employed by each iteration can be seen:

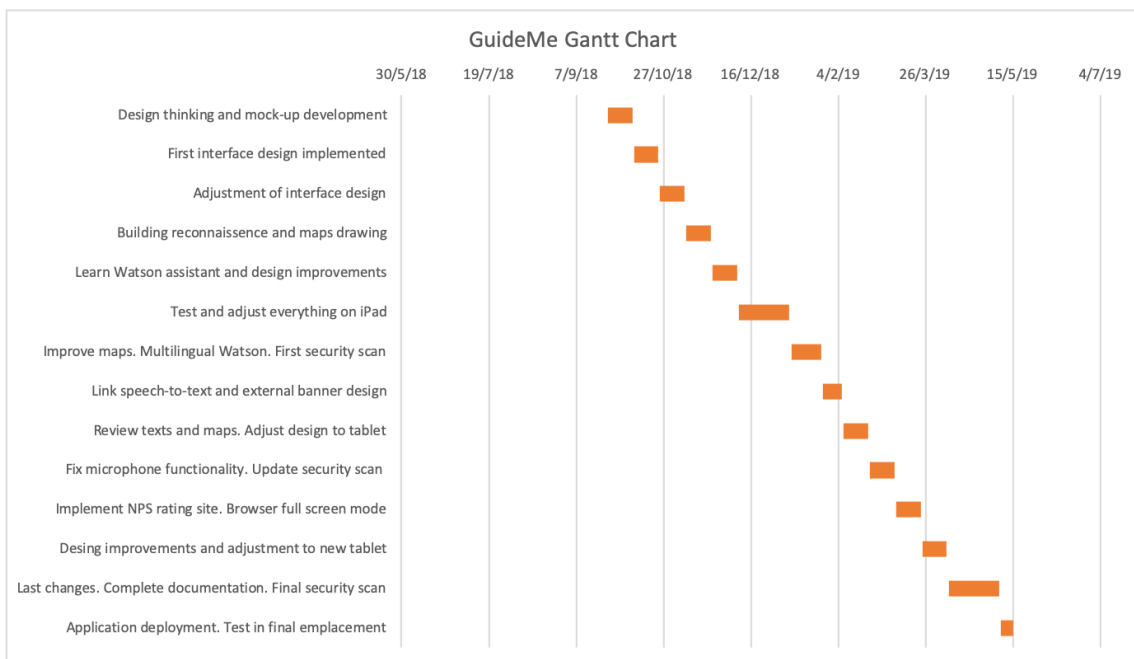


Fig. 5.2. GuideMe Gantt chart

As it can be seen, looking at the previous table 5.1 and to this diagram, most iterations have the same duration, fourteen days, or what is the same, two weeks. Nevertheless, some of them have the length of two iterations and others last a bit more or a bit less than

one, being the shorter iteration length half a normal one, which was the last iteration of the project.

There are several causes which implied the modification of iterations' length. Some iterations matched periods of bank holidays, so iteration real work days were not enough to finish the normal workload. Therefore, their length had to be expanded so they could cover the normal workload due to this lack of work days.

Other iterations were cornerstones, which means that until the iteration scheduled workload was not finished, no new tasks could be begun (E.g. Implementing IBM *Watson* in GuideMe).

The final iteration length is half an iteration since all remaining tasks could be completed within a week and GuideMe release needed to be performed as soon as possible.

### **5.3. Socio-Economic Environment**

GuideMe development has provided IBM a new indoor guidance system to locate most of the rooms and facilities inside the building.

This application has been specifically done for IBM Madrid building, but due to code structure, modifying just certain classes and files will allow an application adjustment to different buildings.

Maps have been specifically drawn for this scenario, but they can be modified following the same nomenclature, and the functions included in *MapValue.js* file can be modified too in order to adjust their behaviour to different environments.

All of this means that GuideMe is not an entirely closed application, and by performing some changes in just a couple of archives it can be used in different buildings, so it can be exported outside IBM as a simple location and guidance system for indoor environments.

## 6. CONCLUSIONS

### 6.1. Objectives Met

Once the application development was concluded and it was installed at IBM building hall, GuideMe was completely operative.

To measure the application effectiveness and performance, the main tool used was the *NPS* scale and user comments and suggestions, obtained from the dedicated space inside the application and from direct questions to those who have interacted with the application.

*NPS* scores and comments are sent to a database, and these data are completely anonymous, being only possible to know the feedback but not who has written it. A few days after the stand was deployed in the main hall, the following data were recorded in the internal database.

	123 SCORE 	ABC COMMENTS 
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	como mola!
7	10	
8	5	Muy bien
9	10	

Fig. 6.1. *NPS* outcome

As it can be seen, *NPS* score is mostly favorable, being only one score different from 10 at the moment. Nevertheless, there are only 2 comments which can be analyzed, and the feedback provided by them only proves the average rating of the solution, so it cannot be used to modify the application so it can be better adjusted to users' needs and to solve

any kind of issue.

Therefore, taking into account the scoring and direct feedback received in the interview published in IBM internal networks in form of comments, it is possible to say that, at the moment, the application has fulfilled the expectations and that it behaves and performs its task properly.

## **6.2. Future Lines of Work**

When the *Design Thinking* meeting was arranged, several potential features of GuideMe were proposed.

As mentioned above, *Design Thinking* has as one of its main basis *brainstorming*, which implies that many different ideas are proposed and that only some of them are finally implemented in the solution. But it does not mean that all ideas which are not originally implemented are discarded.

The first implemented ideas are those which provide more value to the solution and which are better in the ratio priority-cost. Nevertheless, some ideas also provide important value to the application, but they are not of the same priority as others, or their cost is too high to be implemented at the beginning.

So now that GuideMe is finally up and running, it needs continuous technical support to be up-to-date and some of those previous ideas which were devised but not included in the first version of the application can be now implemented.

Some of those ideas were:

- Include beacons inside the building, which provide several location points, meaning that it is not necessary to start all instructions and guidance from the IBM hall where GuideMe stand is located.
- Design a proper mobile interface for personal smartphones, since in the first version the interface was optimized for tablet use, but not intended to be used on a mobile phone. This leads to the next item.
- Mobile phone access for all IBM employees, making possible to use the application from personal smartphones and not being limited just by the GuideMe stand.

- Spread QR codes around the building, allowing potential users to access the application just by scanning the code instead of using the GuideMe stand.

These ideas will be possibly included in future versions of the application. Moreover, the CIO department, which was the one who developed GuideMe, is in charge of developing solutions to ease IBM workers working days. Therefore, GuideMe is not the only application developed by the team. More applications and solutions have been developed as well, related to different needs IBM workers may have, such as getting information about the services provided by IBM, cafeteria working hours..., as well as those related to specific facilities, such as booking a determined room or a car park.

Therefore, the main plan IBM CIO has in mind is to develop a common application which integrates each one of the already developed services, providing IBM employees a universal application which will allow them to interact with all services IBM provides inside the building.

This application will integrate all developed services and will interact with them by means of different *APIs* eventually developed. This will allow, from a single IBM *Watson* service, looking for a room (GuideMe), asking for information about a specific service provided by IBM, booking a room, a car park...

Summing up, IBM CIO plans to develop a universal application which integrates all previously and future developed services, providing IBM employees a global toolkit for all IBM services in Spain.

## ANNEX: REQUIREMENTS

### Functional Requirements

ID	GUIDEME-FR-FUNC-001
TITLE	Start Over
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-FUNC-001
SUBSYSTEM	Functions
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall permit users to start over and look for another room
TEST PLAN	Check if it is possible to go back to the main page once instructions have been displayed

Table 6.1. GUIDEME-FR-FUNC-001

ID	GUIDEME-FR-FUNC-002
TITLE	Photo Reminder
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-FUNC-002
SUBSYSTEM	Functions
TYPE	Functional
PRIORITY	Low
STATUS	Approved
DESCRIPTION	The system shall inform the user about taking a picture of the route
TEST PLAN	Check if a reminder warning appears in the last instruction during 5 seconds

Table 6.2. GUIDEME-FR-FUNC-002

ID	GUIDEME-FR-FUNC-003
TITLE	Auto-Refresh
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-FUNC-003
SUBSYSTEM	Functions
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall auto reload to the main page after two minutes
TEST PLAN	Check if after two minutes without interacting with the application it reloads to the main page

Table 6.3. GUIDEME-FR-FUNC-003



ID	GUIDEME-FR-FUNC-004
TITLE	Micro Reminder
AUTHOR	IBM CIO
CREATION DATE	07-03-2019
SOURCE	GUIDEME-FR-FUNC-004
SUBSYSTEM	Functions
TYPE	Functional
PRIORITY	Low
STATUS	Approved
DESCRIPTION	The system shall inform the user about the possibility of using the microphone
TEST PLAN	Check that a microphone reminder appears in the main page during 5 seconds

Table 6.4. GUIDEME-FR-FUNC-004

ID	GUIDEME-FR-INST-001
TITLE	Instruction System
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-INST-001
SUBSYSTEM	Instructions
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall show the instructions for the target room
TEST PLAN	Check that specific instructions appears when a valid room is processed

Table 6.5. GUIDEME-FR-INST-001

ID	GUIDEME-FR-INST-002
TITLE	Instructions Steps
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-INST-001
SUBSYSTEM	Instructions
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall show the complete route in two steps or less
TEST PLAN	Check that the maximum number of steps shown to get to any room is two or less

Table 6.6. GUIDEME-FR-INST-002

ID	GUIDEME-FR-INP-001
TITLE	Multiple Input Options
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-INP-001
SUBSYSTEM	Input
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall allow to introduce the input in multiple ways
TEST PLAN	Check that there are several working ways to introduce the target room

Table 6.7. GUIDEME-FR-INP-001

ID	GUIDEME-FR-INP-002
TITLE	Written Input
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-INP-001
SUBSYSTEM	Input
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall permit users to introduce the desired input via typing on a keyboard
TEST PLAN	Check if it is possible to introduce a room via typing in the main text box

Table 6.8. GUIDEME-FR-INP-002

ID	GUIDEME-FR-INP-003
TITLE	Voice Input
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-INP-001
SUBSYSTEM	Input
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall permit users to introduce the desired input via voice commands
TEST PLAN	Check if it is possible to introduce a room via voice commands pressing the microphone icon

Table 6.9. GUIDEME-FR-INP-003

ID	GUIDEME-FR-INP-004
TITLE	Recognition Of Typed Natural Language
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-INP-001
SUBSYSTEM	Input
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall recognise a specific room from a typed natural language sentence
TEST PLAN	Check if the system recognises the target room referred to in a typed natural language sentence

Table 6.10. GUIDEME-FR-INP-004

ID	GUIDEME-FR-INP-005
TITLE	Recognition Of Spoken Natural Language
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-INP-001
SUBSYSTEM	Input
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall recognise a specific room from a spoken natural language sentence
TEST PLAN	Check if the system recognises the correct room referred to in a spoken natural language sentence

Table 6.11. GUIDEME-FR-INP-005

ID	GUIDEME-FR-NPS-001
TITLE	NPS System
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-NPS-001
SUBSYSTEM	NPS
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall implement a NPS rating system
TEST PLAN	Check that NPS rating site works within the application

Table 6.12. GUIDEME-FR-NPS-001

ID	GUIDEME-FR-NPS-002
TITLE	NPS Database
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-NPS-001
SUBSYSTEM	NPS
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall implement a database to record users' ratings
TEST PLAN	Test the connection of the database to the application with test ratings

Table 6.13. GUIDEME-FR-NPS-002

ID	GUIDEME-FR-NPS-003
TITLE	Opinions Database
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-NPS-001
SUBSYSTEM	NPS
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall implement a database to record users' opinions about GuideMe
TEST PLAN	Test the connection of the database to the application with test opinions

Table 6.14. GUIDEME-FR-NPS-003

ID	GUIDEME-FR-WAT-001
TITLE	Multilingual Watson
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-WAT-001
SUBSYSTEM	Watson
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	IBM <i>Watson</i> needs to be implemented in several languages
TEST PLAN	Test individually each IBM <i>Watson</i> implementation request and response displayed language

Table 6.15. GUIDEME-FR-WAT-001

ID	GUIDEME-FR-WAT-002
TITLE	English Watson
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-WAT-001
SUBSYSTEM	Watson
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	IBM <i>Watson</i> needs to be implemented in English
TEST PLAN	Check IBM <i>Watson</i> request and response are displayed in English

Table 6.16. GUIDEME-FR-WAT-002

ID	GUIDEME-FR-WAT-003
TITLE	Spanish Watson
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-WAT-001
SUBSYSTEM	Watson
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	IBM <i>Watson</i> needs to be implemented in Spanish
TEST PLAN	Check IBM <i>Watson</i> request and response are displayed in Spanish

Table 6.17. GUIDEME-FR-WAT-003

ID	GUIDEME-FR-WAT-004
TITLE	Written English Watson
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-WAT-001
SUBSYSTEM	Watson
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	IBM <i>Watson</i> needs to understand written English inputs
TEST PLAN	Introduce valid inputs in English and check that the response is the proper one

Table 6.18. GUIDEME-FR-WAT-004

ID	GUIDEME-FR-WAT-005
TITLE	Written Spanish Watson
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-WAT-001
SUBSYSTEM	Watson
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	IBM <i>Watson</i> needs to understand written Spanish inputs
TEST PLAN	Introduce valid inputs in Spanish and check that the response is the proper one

Table 6.19. GUIDEME-FR-WAT-005



ID	GUIDEME-FR-WAT-006
TITLE	Spoken English Watson
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-WAT-001
SUBSYSTEM	Watson
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	IBM <i>Watson</i> needs to understand spoken English inputs
TEST PLAN	Introduce valid spoken inputs in English and check that the response is the proper one

Table 6.20. GUIDEME-FR-WAT-006

ID	GUIDEME-FR-WAT-007
TITLE	Spoken Spanish Watson
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-FR-WAT-001
SUBSYSTEM	Watson
TYPE	Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	IBM <i>Watson</i> needs to understand spoken Spanish inputs
TEST PLAN	Introduce valid spoken inputs in Spanish and check that the response is the proper one

Table 6.21. GUIDEME-FR-WAT-007

# Consistency Matrix

	FUNC-001	FUNC-002	FUNC-003	FUNC-004	INST-001	INST-002	INP-001	INP-002	INP-003	INP-004	INP-005	NPS-001	NPS-002	NPS-003	WAT-001	WAT-002	WAT-003	WAT-004	WAT-005	WAT-006	WAT-007	
FUNC-001																						
FUNC-002																						
FUNC-003																						
FUNC-004																						
INST-001																						
INST-002																						
INP-001																						
INP-002																						
INP-003																						
INP-004																						
INP-005																						
NPS-001																						
NPS-002																						
NPS-003																						
WAT-001																						
WAT-002																						
WAT-003																						
WAT-004																						
WAT-005																						
WAT-006																						
WAT-007																						

Fig. 6.2. Functional requirements consistency matrix

## Non-Functional Requirements

ID	GUIDEME-NF-INST-001
TITLE	Instructions Size
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-NF-INST-001
SUBSYSTEM	Instructions
TYPE	Non-Functional
PRIORITY	Medium
STATUS	Approved
DESCRIPTION	The instructions have to be of no more than two lines
TEST PLAN	Check that instructions' size never is bigger than two lines

Table 6.22. GUIDEME-NF-INST-001

ID	GUIDEME-NF-INST-002
TITLE	Instructions Comprehension
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-NF-INST-001
SUBSYSTEM	Instructions
TYPE	Non-Functional
PRIORITY	Medium
STATUS	Approved
DESCRIPTION	The instructions shall be <i>WAI AA</i> compliant
TEST PLAN	Check that instructions are standard compliant

Table 6.23. GUIDEME-NF-INST-002

ID	GUIDEME-NF-INST-003
TITLE	Successful Input Message
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-NF-INST-003
SUBSYSTEM	Instructions
TYPE	Non-Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall return a successful input message once a valid room is processed
TEST PLAN	Check if the system returns a success message when introducing an existing room

Table 6.24. GUIDEME-NF-INST-003

ID	GUIDEME-NF-INST-004
TITLE	Unsuccessful Input Message
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-NF-INST-004
SUBSYSTEM	Instructions
TYPE	Non-Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall return an error input message once a non-valid room is processed
TEST PLAN	Check if the system returns an error message when introducing an nonexistent room

Table 6.25. GUIDEME-NF-INST-004

ID	GUIDEME-NF-INST-005
TITLE	Successful Input Instructions
AUTHOR	IBM CIO
CREATION DATE	07-03-2019
SOURCE	GUIDEME-NF-INST-005
SUBSYSTEM	Instructions
TYPE	Non-Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall load the instructions first step after processing a valid room
TEST PLAN	Check if the system loads the next page after introducing a valid room

Table 6.26. GUIDEME-NF-INST-005

ID	GUIDEME-NF-INST-006
TITLE	Unsuccessful Input Instructions
AUTHOR	IBM CIO
CREATION DATE	07-03-2019
SOURCE	GUIDEME-NF-INST-006
SUBSYSTEM	Instructions
TYPE	Non-Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The system shall not load any instructions or load an error page after introducing a non-valid room
TEST PLAN	Check if the system loads no page or an error page only

Table 6.27. GUIDEME-NF-INST-006

ID	GUIDEME-NF-DESIGN-001
TITLE	Simple Design
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-NF-DESIGN-001
SUBSYSTEM	Design
TYPE	Non-Functional
PRIORITY	Low
STATUS	Approved
DESCRIPTION	The application design has to be simple
TEST PLAN	Check that the design does not overwhelm the user

Table 6.28. GUIDEME-NF-DESIGN-001

ID	GUIDEME-NF-NPS-001
TITLE	NPS Rating
AUTHOR	IBM CIO
CREATION DATE	07-03-2018
SOURCE	GUIDEME-NF-NPS-001
SUBSYSTEM	Security
TYPE	Non-Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The users have to be able to rate the application via NPS rating
TEST PLAN	Check that users' ratings are visible into a specific database

Table 6.29. GUIDEME-NF-NPS-001

ID	GUIDEME-NF-NPS-002
TITLE	Users' Opinions
AUTHOR	IBM CIO
CREATION DATE	07-03-2018
SOURCE	GUIDEME-NF-NPS-001
SUBSYSTEM	Security
TYPE	Non-Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The users have to be able to leave opinions about the application when rating
TEST PLAN	Check that users' opinions are visible into a specific database

Table 6.30. GUIDEME-NF-NPS-002

ID	GUIDEME-NF-GDPR-001
TITLE	GDPR Compliant
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-NF-GDPR-001
SUBSYSTEM	GDPR
TYPE	Non-Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The application has to be compliant with the GDPR law
TEST PLAN	Application GDPR analysis providing a successful outcome

Table 6.31. GUIDEME-NF-GDPR-001

ID	GUIDEME-NF-SEC-001
TITLE	Security Compliant
AUTHOR	IBM CIO
CREATION DATE	25-09-2018
SOURCE	GUIDEME-NF-SEC-001
SUBSYSTEM	Security
TYPE	Non-Functional
PRIORITY	High
STATUS	Approved
DESCRIPTION	The application has to be compliant with the IBM security standards
TEST PLAN	Application IBM Security analysis providing a successful outcome

Table 6.32. GUIDEME-NF-SEC-001



## BIBLIOGRAPHY

- [1] *Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales*, Available at <https://www.boe.es/eli/es/lo/2018/12/05/3>.
- [2] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, “Indoor localization without the pain”, in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, ACM, 2010, pp. 173–184.
- [3] J. Chung *et al.*, “Indoor location sensing using geo-magnetism”, in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, ACM, 2011, pp. 141–154.
- [4] M. Campbell, A. J. Hoane Jr, and F.-h. Hsu, “Deep blue”, *Artificial intelligence*, vol. 134, no. 1-2, pp. 57–83, 2002.
- [5] F.-h. Hsu, “Ibm’s deep blue chess grandmaster chips”, *IEEE Micro*, vol. 19, no. 2, pp. 70–81, 1999.
- [6] A. J. Ko, *A history of user interfaces*, Available at <https://faculty.washington.edu/ajko/books/uist/history.html>.
- [7] A. Hopkins, *An abridged history of ui*, Available at <https://blog.prototypr.io/an-abridged-history-of-ui-7a1d6ce4a324> (Feb 2, 2017).
- [8] J. Markoff, *It’s moore’s law, but another had the idea first*, Available at <https://nyti.ms/21jK90u> (Apr. 18, 2005).
- [9] D. Norman, *The design of everyday things: Revised and expanded edition*. Basic books, 2013.
- [10] S. Pillai, *View from the labs: Voice user interfaces – a short history and a bright future*, Available at <https://www.infostretch.com/blog/view-from-the-labs-voice-user-interfaces-a-short-history-and-a-bright-future/> (Dec. 8, 2016).
- [11] R. A. Bolt, “Put-that-there”: *Voice and gesture at the graphics interface*, 3. ACM, 1980, vol. 14.

- [12] D. H. Rose and A. Meyer, *Teaching every student in the digital age: Universal design for learning*. ERIC, 2002.
- [13] B. Shneiderman and C. Plaisant, *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education India, 2010.
- [14] D. H. Klatt, “Review of text-to-speech conversion for english”, *The Journal of the Acoustical Society of America*, vol. 82, no. 3, pp. 737–793, 1987.
- [15] Y. Zhong, T. Raman, C. Burkhardt, F. Biadysy, and J. P. Bigham, “Justspeak: Enabling universal voice control on android”, in *Proceedings of the 11th Web for All Conference*, ACM, 2014, p. 36.
- [16] A. Rodrigues, K. Montague, H. Nicolau, and T. Guerreiro, “Getting smartphones to talkback: Understanding the smartphone adoption process of blind users”, in *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, ACM, 2015, pp. 23–32.
- [17] D. Bohn, *Amazon says 100 million alexa devices have been sold — what’s next?*, Available at <https://www.theverge.com/2019/1/4/18168565/amazon-alexa-devices-how-many-sold-number-100-million-dave-limp>.
- [18] T. Brown, “Design thinking”, *Harvard business review*, vol. 86, no. 6, p. 84, 2008.
- [19] *What is brainstorming*, Available at <https://businessjargons.com/brainstorming.html>.
- [20] *A beginners guide to understanding the agile method*, Available at <https://project320.com/resources/the-agile-method/>.
- [21] J. S. Ken Schwaber, *The scrum guide*, Available at <https://www.scrumguides.org/scrum-guide.html> (2017).
- [22] R. High, “The era of cognitive systems: An inside look at ibm watson and how it works”, *IBM Corporation, Redbooks*, 2012.
- [23] Y. Chen, J. E. Argentinis, and G. Weber, “Ibm watson: How cognitive computing can be applied to big data challenges in life sciences research”, *Clinical therapeutics*, vol. 38, no. 4, pp. 688–701, 2016.
- [24] *What is net promoter?*, Available at <https://www.netpromoter.com/know/>.

