Carolina Ornelas Lopes Caria da Silva

Licenciada em Ciências da Engenharia
Eletrotécnica e de Computadores

Mobile Application to Support Intelligent Supervision System for Service Buildings

Dissertação para obtenção do Grau de Mestre em
Engenharia Eletrotécnica e de Computadores

Orientador: Doutora Ana Inês da Silva Oliveira, Professora Auxiliar,
Faculdade de Ciências e Tecnologia da
Universidade NOVA de Lisboa

Fevereiro, 2021

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Mobile Application to Support Intelligent Supervision System for Service Buildings

Acknowledgments

First of all, I want to thank NOVA FCT and all my teachers for welcoming me into this institution, for teaching me with patience and devotion in the various areas included in this big course.

I want to give a big thank you to Prof. Ana Inês Oliveira for all the support and advises she gave me and without which this work would never see the light of day.

Thank you to Prof. João Rosas for the advises he gave me in the early stages of this dissertation work.

Big thank you to my mother and family for all the support not only in these five years but throughout my entire life. Without you I never would have gotten this far.

Thank you to my friends, the ones I had before starting this journey and stuck with me till the end, and the ones that I met during this time. You all know how important you are.

To you, for always being there with me, listening to me, encouraging me in my bad moments, thank you.

Thank you very much to you all!

Abstract

The work developed and described in this dissertation is part of the Ambiosensing project, developed under the Portugal 2020 program. This project aims to design and develop a tool for the energy management of buildings, considering low implementation costs, adaptability, versatility, and easy maintenance in line with the premises of Industry 4.0. One of the main requirements of the project is related to the intelligent supervision of equipment, adaptability and optimization of energy efficiency and quality of comfort of the occupants of buildings.

In this way, the problem that this dissertation addresses is related to the comfort of the occupants within a service building. For that purpose, an application for mobile devices was designed and developed complementing the Intelligent Supervision system developed in the project. This application makes it possible to view the values of the registered environmental variables and allows the users of the spaces to leave their feedback regarding their feeling considering the presented values, in order to improve the performance of the supervision system. In addition to allowing the connection between the user and the system improving not only the system's performance, but the application also improves the user's experience inside the building.

Keywords: Energy management, Control system, Intelligent supervision, Application for control system

Resumo

O trabalho desenvolvido e descrito nesta dissertação está integrado no projeto Ambiosensing, desenvolvido no âmbito do programa Portugal 2020. Este projeto tem como objectivo a concepção e desenvolvimento de uma ferramenta para a gestão energética de edifícios, considerando baixos custos de implementação, adaptabilidade, versatilidade e fácil manutenção alinhado com as premissas da Indústria 4.0. Um dos principais requisitos do projecto está relacionado com a supervisão inteligente dos equipamentos, adaptabilidade e optimização de eficiência energética e qualidade de conforto dos ocupantes dos edifícios.

Desta forma, o problema que esta dissertação aborda está relacionado com o conforto dos ocupantes dentro de um edifício de serviços e para tal foi desenhada e desenvolvida uma aplicação para dispositivos moveis que serve de complemento ao sistema de Supervisão Inteligente desenvolvido no projecto. Esta aplicação possibilita a visualização dos valores das variáveis ambientais registados permite que os utilizadores dos espaços deixem o seu feedback em relação à sua semsibilidade sobre os valores apresentados, com o intuito de melhorar a performance do sistema de supervisão. Além de permitir a ligação entre o utilizador e o sistema melhorando, não só a performance do mesmo, a aplicação permite também melhorar a experiência do utilizador no interior do edificio.

Palavras-chave: Gestão de energia, sistemas de controlo, Supervisão Inteligente, Aplicação para sistema de controlo.

List of Contents

List of Figures

List of Tables

List of Acronyms

ANN	Artificial Neural Network
BAS	Building Automation System
BEMS	Building Energy Management System
BMS	Building Management System
DBMS	Database Management System
EMS	Energy Management System
FLC	Fuzzy Logic Controllers
Fuzzy-PD	Fuzzy-Proportional Derivative
FSGIM	Facility Smart Grid Information Models
GA	Genetic Algorithm
GUI	Graphical User Interface
HVAC	Heat, Ventilation and Ari Conditioning
IAQ	Indoor Air Quality
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
LED	Light Emitting Diode
PID	Proportional Integrative Derivative
RUP	Rational Unified Process
SA	Simulated Annealing
SQL	Structure Query Language
SVM	Support Vector Machines
UI	User Interface
UML	Unified Modeling Language

1. Introduction

1.1 Problem Domain and Motivation

People spend most of their lives inside buildings, so the indoor environment is an important aspect that should be ideal to provide its users with the comfort they need.

A service building can be characterized as a building that offers services (can be a public or private services or commercial) and can include systems installed to make it more functional, efficient and safe. These systems may include water, drainage and plumbing, Building management systems, escalators and lifts, security and alarm systems, etc. [1]

Therefore, service buildings occupants' comfort can be affected by a great number of different factors such as [2]:

Personal factors: include the persons age, level of health, the clothes he/she is wearing at the time and the type of activity that person is doing while inside the building;

Thermal comfort: "That condition of mind which expresses satisfaction with the thermal environment and is assessed by subjective evaluation" according to [3]. A good thermal environment it is always imperative inside a building as it can affect the people's health and have an impact on their ability to make their activities effectively;

Indoor air quality: users health and wellbeing can be greatly affected by the capacity of a building to ventilate the air inside. Ventilation is necessary to replace the air inside the building with fresh air, this air motion can have a great impact in the Air quality inside [4];

Visual comfort: the intensity of light either to low or to high can be very uncomfortable. This factor can be addressed by controlling the natural light that comes into the building and the intensity of the bulbs that give artificial light to a room [4].

In order to improve occupants' comfort levels, considering the fact that the population is growing, and the fact that people tend to spend more time indoors than outdoors, energy consumption in service buildings was raised to the level of industry and transport [5] .

Therefore, a way to prevent the energy consumption from increasing even more is to invest in "Intelligent Buildings".

A "Intelligent Building" is able to adapt to its occupants needs, fluctuating situations and past requirements, while having the goal to increase the energy efficiency of the building [6].

"Energy Efficiency refers to using less energy to produce the same services or useful output"[7]. Which means, "Intelligent Buildings" are able to save up energy without compromising the comfort inside and affect negatively the productivity of its occupants.

Considering the service building context, to keep an energy efficient environment, it should be possible to monitor, in real time, occupancy and user context, environmental conditions and energy usage of the space [7].

Occupancy and user context refer to how many people are in a room which activity they are performing in the space. Having the Heat, Ventilation and Air

Conditioning (HVAC) system, for instance, ON in an empty room would be a waste of energy. Moreover, having the HVAC system ventilating the space considering the room full occupancy when it is not the case, it's not very efficient either.

To prevent energy waste and reduce the costs associated with energy consumption it is necessary to implement solutions that will increase the energy efficiency of the building while keeping the occupants' comfort and the environment quality inside. The solution presented in this dissertation work, supports some of these objectives with the development of a mobile application to support an intelligent supervision system, aiming at monitoring the occupants' comfort and collecting their opinion/feedback.

1.2 Ambiosensing

This dissertation work was carried out within the scope of the Ambiosensing research project [8], financed by the Portugal 2020 program.

The Ambiosensing (Autonomous & Intelligent System for Energy Saving) is a project focused on the topic of energy efficiency in buildings. Its primary goal is to develop low-cost implementation solution for the management of energy and air quality in service buildings.

The developed solution in the Ambiosensing project must be able to communicate with previously installed technology in the buildings incorporating IoT technology and "machine learning" while having the capacity to manage Big Data. So, the building must be prepared with necessary hardware for this project to be implemented, like an Intelligent Building.

The Ambiosensing project will work to detect the main energy efficiency flaws in the building and intervene to rectify those flaws in the equipment that is contributing for the building's poor energy management., without compromising the occupants' comfort and the environmental quality inside the building.

1.3 Objectives and Contributions

Considering the indoor occupants' comfort, this dissertation work aims to contribute to the development of a supervision system intended to monitoring the functioning and condition of buildings, in order to improve the environmental quality inside service buildings assuring the comfort of its users while preventing energy waste and reducing costs associated with energy consumption. More specifically, this work aims at designing and developing a mobile application that, considering the monitoring of environment variables, provides its values to the building occupants and collects their feedback on their comfort feeling. This feedback is then used by the supervision system to take any corrective or adjustive measure.

As previously explained, this dissertation work was carried out within the scope of the Ambiosensing research project, and some of the main objectives of this project are:

- Research and development of optimization tactics applied to the system to be developed, providing a reduction in operating costs and resulting in real energy savings.
- Integration of the analysis of local Indoor Air Quality (IAQ) conditions and automatic adjustment of the elements that control it (fan motors, etc.), with learning capabilities that aim to reduce the reaction time of the system and the adjustment to desired conditions with minimum energy consumption.
- Adoption of management practices and reduction of carbon emissions, with a focus on research and development of algorithms and techniques that potentiate decarbonization.
- System stability, with the ability to connect to existing networks, with a high number of established protocols and the ability to connect and transmit data over Wi-Fi networks, its own router and /or Internet of Things (IoT).

In order to provide users with data and information about the state of the room they are in (room temperature, air quality, etc.), and allow an interaction between the user and the Supervision system an application for mobile devices, is implemented. This application can also receive feedback from building occupants regarding how comfortable they feel in the space. This dissertation work focuses on the development of this application that works as a connection between the buildings' occupants and the supervision system allowing the users to participate in the process of improving the supervision system performance while visualizing the environmental data that characterizes the space.

1.4 Structure of the Document

In the next chapter, the study of the conceptualization and literature summary is exposed with some of the scientific and technological directions that exist within the theme of this dissertation work.

The third chapter presents the application designing process, where some development methodologies were considered. The application requirements and UML diagrams are also presented.

The fourth chapter goes into detail on how the application was developed explaining the process from the selection of the development tools to the application implementation.

The fifth chapter presents the testing and validation phase, where some of the tests and scenario of use is described.

Finally, section six presents this work conclusions and possible future work and improvements.

2. Background and Literature Review

2.1 Intelligent Buildings

The term "Intelligent Building" was introduced in the 1980s and since then has had several definitions. The first one, given by the Intelligent Buildings Institute, defined Intelligent Building as "one which provides a productive and cost-effective environment through optimization of four basic elements: structure, systems, services and management, and the interrelationship between them."[9]. Which means that an Intelligent Building optimally matches its subsystems to increase users' comfort.

In the 1990s and early 2000s, Intelligent Buildings definitions started to be tilted towards energy efficiency and sustainability and in the late 2000s the term "Bright Green Buildings" was mentioned by Frost and Sullivan's research paper commissioned by the Continental Automated Buildings Association [10].

More recently, definitions have into account the Internet of Things (IoT) impact in technology referring that the building services are connected to monitor, analyze and control the building without human intervention [9].

Even with all the different definitions, it's clear that an Intelligent Building is a building with a connective platform that controls various subsystems inside the building (HVAC, lightning, etc.) with the goal to improve energy efficiency, space utilization and occupant satisfaction.

2.2 Energy Management Systems

An Energy Management System (EMS) is a system of computational tools used to monitor, control and optimize the energy efficiency of a system.

Building Energy Management System (BEMS) is part of a more complex monitoring system that monitors and controls the mechanical and electronic equipment of the building, such as lighting, HVAC, security systems, etc [11].

These systems are capable of monitoring and metering systems of a building to collect energy data allowing the owners insight on the building's energy usage, contributing to the building's energy efficiency.

BEMS work by interacting components and systems already existing in the building. They usually communicate with two categories of building systems: Infrastructures and Building Software [11].

Existing Infrastructure in the building refers to physical systems and sensors, such as heaters, elevators, security systems, etc. When connected to these Infrastructures, BEMS is able to monitor the energy consumption of these physical systems.

Building Software refers to Building Management Systems (BMS) and Building Automation Systems (BAS). BMS helps to manage different parts of the building while BAS is more focused on automating building management.

These connections allow BEMS to collect and analyze data from all systems, ensuring a more accurate understanding of energy use.

In addition to the objective of reducing the energy consumption of a building, a component that cannot be ignored is comfort. Comfort plays an essential role in preserving health, morals, efficiency at work, productivity and user satisfaction [6].

Considering the above, any system should create a balance between energy efficiency and comfort, thus achieving the objective of reducing energy consumption and CO_2 emissions, improving the environmental comfort inside the building.

2.3 Control Systems

A solution that has been increasingly used not only to be able to control the energy consumption of a building, but also the comfort and environmental status of each space inside, are the control systems. Control systems can manage and control systems according to a defined objective and following a set of rules.

A controller is a device that receives information about the environment and produces responses that manipulate the state of the environment [12].

For the control and monitoring within a building to be possible, it is necessary to invest in hardware equipment. This equipment is interconnected to form the control system[13] and can be:

- **Sensors:** the equipment responsible for monitoring and obtaining real-time information about the state of the room that is being controlled. Various types of sensors can be installed for a more sophisticated system [14].
  To detect the temperature, using the example above, temperature sensors would be necessary for the controller to obtain this information. Other types of sensors such as light sensors for lighting control, smoke sensors for fire detection, sensors for controlling the occupation of a room, etc. are also important for the control of an intelligent building.
- **Actuators:** the devices that will act according to what the controller tells them. Examples of such devices are switches, motors, etc.
- **Controllers**: control the equipment taking into account previously established rules and restrictions [13].

Interface devices refers to devices that allow interaction between the user and the control system. With these devices they allow the user to be able to model the room environment for their comfort [15].

There are several solutions that can be implemented in buildings to increase energy efficiency. Some examples of the implementation of these solutions are described in the next subsections.

2.3.1 Lighting Control

Lighting is one of the main systems that can affect a building's energy consumption. Having good control under lighting systems that aims to reduce energy consumption is essential to reduce the energy costs of a building.

Since this project objective is to reduce energy costs without affecting comfort inside the building, the lighting level in a room must be adequate considering the activity that is practiced in the room. If it is possible to achieve this level with natural lighting during the day, it is not necessary to spend energy using artificial lighting.

It is also possible to save energy by replacing incandescent lamps with LED (light-emitting diode) lamps, these lamps can save up to 75% more energy and last 25 times longer [16].

In [17] there is a solution for lighting control that allows users to directly control the lighting of the space. These manual regulators are not very energy efficient as the user has no way of knowing the effect that their control has on energy efficiency.

An efficient way to control lighting would be considering the occupants of the room. If the room is empty, the system must turn off the space lighting to save energy [13].

2.3.2 Plug-load Control

Plug-load devices such as monitors, or chargers are often left ON when they are no longer in use. Due to the large number of such devices in a building, the aggregate of all this waste of energy is quite significant.

The easiest way to control the costs on this type of device would be human control, which would consist only of turning off the device when it is no longer in use.

In [13] a plug-load meter is presented that can disconnect the electricity and turn off the device, the plug-load meter would be controlled according to the occupancy of the space.

In [18] there is an example of Smart Plugs and Power Meters. With these smart plugs the user can control, from an application, the devices being able to disconnect them using the application on your Smartphone or computer. It is also possible for the user to have access to the power consumption of the devices.

2.3.3 HVAC Control

HVAC (Heat, Ventilation and Air Conditioning) systems are responsible for controlling the heating, air quality and humidity inside a building, with the objective to provide comfort and the best productive environment. However, these systems are responsible for the consumption of almost 50% of building's energy (Afram & Janabi-Sharifi, 2014). Therefore, a good HVAC control, is crucial to increase a building's energy efficiency [15], [19].

The HVAC systems can control the temperature in a room, so a desired comfort temperature is set for the system to reach, this reference temperature is called Set Point. Different rooms in a building can have different set points that can be adjusted to fit the occupant's preferences [20].

Heat, Ventilation and Air Conditioning systems consume more energy if they have to provoke a sudden change in a room's temperature. To avoid this, a way to preserve energy would be making the temperature change gradually instead of suddenly [15].

In [21] a solution is presented which consist in scheduling features in the HVAC system so that the desired temperature of the room can be achieved gradually, saving energy, and when the occupants arrive, the room would already be at the desired temperature.

Another problem with HVAC systems nowadays is that they are programed to condition a room considering the maximum occupancy of the room, instead of the number of people that are actually in the space [22].

In [22] sensors are used to gather data about the number of people in a room. This data is then used to create an occupancy model that can be implemented in the conditioning strategies of the building. This model is then used to predict when a room will be occupied so that the HVAC can start conditioning the room before its occupied.

In [23] there is a review of different HVAC control methods. This control methods can be divided into "Local control" and "Supervisory control".

Local control allows the building systems to operate with basic control. This kind of control can be energy efficient when certain subsystems perform are concern but not when the overall system is put into consideration [23].

Supervisor control takes into account the indoor and outdoor changing conditions and the HVAC system to provide the indoor comfort and healthy environment using the minimum input energy [23].

Contrary to the Local control, Supervisory control allows the overall consideration of the system. The data it gathers can be utilized to improve the system energy efficiency.

2.3.4 Existing Interface Devices for Control Systems

Nowadays it is possible for the user to control the control system of a house or building with mobile devices or applications for smartphones or tablets.

With these applications, the user, in addition to being able to adjust the HVACs, lighting and temperature of a space, can also receive cost reports. This allows the user to have an idea of the expenses and adjust the system according to his preferences [15].

However, when considering commercial buildings, not everyone in the space should have access to the control level. Control systems are previously programmed or trained so that the space has the ideal conditions of comfort for users of the space, taking into account the activity that is carried out in the rooms.

In [24], an application, "TherMOOstat", and a control system were created for the various buildings on the college campus. Users of those buildings, in this case students, teachers, etc., can give their opinion regarding the building's temperature level, if they feel good, if they think it is cold, etc. This feedback is then analyzed, and the system is improved so that the user comfort level is the highest possible.

In [25], a mobile app is presented, this app lets the Facilities Teams of the building manage the HVAC system allowing them to see the users comfort levels the users share in the app, making them know right away if anything needs to be changed.

These were the existing Interface Devices found that presented the most similar goals to the application developed for this dissertation.

This dissertation work focuses on the development of a mobile application for the users to see the environmental data saved by a Supervision System and collect the user's feedback to then improve the Supervision System.

2.4 Intelligent Control Systems

Most HVAC systems nowadays are controlled by on/off and proportional integral derivative (PID) controllers due to the simplicity of this control systems, however, because HVAC systems have certain features that others, apparently similar systems, don’t have [19], these types of controllers are not as efficient as controllers implemented with intelligent systems.

An intelligent system is a system capable of imitating and automate intelligent human behaviors. These features allow the system to be able to learn from the previous users and with that knowledge, automatically manage various equipment and make decisions. This allows a more efficient and automated control over the system.

There are several works that combine different intelligent control systems, such as fuzzy logic controllers and Neural Network controllers as described in [26].

Also, other works suggest a fuzzy logic controller as a solution to achieve indoor thermal comfort and air quality while considering energy and cost efficiency [27]. This controller is compared to a conventional HVAC controller, PID. The fuzzy logic controller shows the best results as it has the ability to make decisions incorporating expert knowledge and is able to deal with multivariant problems.

At [28] a practical application of a fuzzy control system for a HVAC system is presented. This control system has into consideration the temperature and humidity of the room to control the cooling, heating and humidity valves. The simulation results show that fuzzy control systems are capable of controlling a Heating, Ventilation and Air Conditioning system efficiently and economically.

Moreover, other authors describe a fuzzy logic energy management supervision system to help reduce Energy cost and CO_2 emissions using photovoltaic and storage systems [29]. Nevertheless, fuzzy logic can also be used to control the compressed and fan speed of an HVAC system, with the propose of improving comfort in different areas [30].

At [31] a self-tuning PID type fuzzy adaptive controller is presented. This is as auto adaptive controller which uses fuzzy logic control to tune the parameters of a PID controller. This model is compared with a classical PID and a fuzzy-Proportional Derivative (fuzzy-PD) controller and, according to the results presented, is the most effective among the others.

Also, in [32] an Adaptive fuzzy-PD controller is compared to a nonadaptive fuzzy-PD and to an ON/OFF controller. MATLAB and SIMULINK were utilized to simulate the building's thermal comfort response to the fuzzy logic controllers.

Moreover, solutions to tune Fuzzy Logic Controllers (FLCs) are given by [33]. The tunning process is divided in two stages. The first one uses a Genetic Algorithm (GA) to fast obtain acceptable tuning parameters, and the second stage Simulated Annealing (SA) to polish the solutions obtained in the first stage.

Furthermore, several examples of artificial neural networks (ANN) able to predict the building's energy consumption, building's thermal loads, air flow in a naturally ventilated room, between others, are described in [34]. Whereas some studies conclude that the ANN's and Support Vector Machines (SVM) systems were the ones with better results to their effectiveness in solving non-linear problems when reviewing various control models and its capability to predict a building consumption [35].

More works, such as [36] present a Neural Fuzzy Assistant is presented. This Decision Support System uses a Neural Network model to estimate building's energy consumption, considering several relevant parameters like the building's characteristics, occupancy density and ventilation. The results obtained on the Neural Network model are then corrected with fuzzy logic.

Also, at [15] a Fuzzy logic and artificial neural networks controller is implemented to increase energy efficiency of a habitation while keeping the thermal comfort. Artificial Neural Network is used to predict the outside temperature and the Fuzzy Logic is used to determine the thermal power the HVAC should release.

These are just some examples of applications of the Fuzzy Logic and Artificial Neural Network controllers that exist. According to these studies, these intelligent solutions for controlling are very effective methods to increase energy efficiency in buildings.

2.4.1 Future Perspectives on Intelligent Control Systems

There are some studies and research work that consider future perspectives in what concerns intelligent control systems. In [37] an IoT-based communication structure with a standard information model is proposed. This study demonstrates that the architecture based on IoT and Facility Smart Grid Information Models (FSGIMs) can greatly improve the communication between equipment ensuring not only a simpler implementation but also a better management over the integrated energy management systems in a building.

Future work mentioned consists in the implementation of a demand response scheme that would support in real time the price of energy in the IoT platform. And improve the IoT-based energy management platform by developing and implementing an initial energy storage system.

Presented in [38] there is a supervision system and management strategies for large buildings. The system is designed with the aim of maintaining comfort inside and reducing energy consumption. The state of the room is detected by wireless sensors and the BAS is integrated into the BEMS that was developed to improve the performance of the equipment and increase energy efficiency.

Future work mentioned includes optimization of the proposed model, adjustment of the automation system, development of uniform communication between systems and software for retrofit decision.

Also in, [39] an intelligent model that uses a set of rules for energy management in buildings is presented as a decision support system in order to guarantee the desired quality of life levels as well as energy savings.

The system allows monitoring of the consumption of energy, decoding the building's energy knowledge, first, into different rules and then to commands for actuator devices.

According to the results of its pilot application, the operation of this model was satisfactory as it contributed to the improvement of indoor air quality and reduced energy consumption.

These examples show that, even though studies and tests have been performed, there is always room for improvement and theories that have yet to be developed.

The Ambiosensing project main goal is to develop a proof of concept for an energy management and air-quality solution. This solution includes an Intelligent Supervision system, which is described in the next section.

2.5 Intelligent Supervision System

An intelligent supervisor in the control system allows, as the name implies, to supervise the control system. Its functions are to coordinate the execution of a plan, detect errors in the system and find solutions or explanations for them.

For example, a room with a heater and a fan, if the control system detects that the temperature in the room is below the desired level, it sends the command to start the heater. The supervision system will supervise the space and see if the temperature increases as it is supposed to since the heater was turned on.

A system architecture is the model that defines the structure and behavior of a system. With an architecture we can have a generalized view of how the system works, its components and how they are interconnected between them and the outside environment [40].

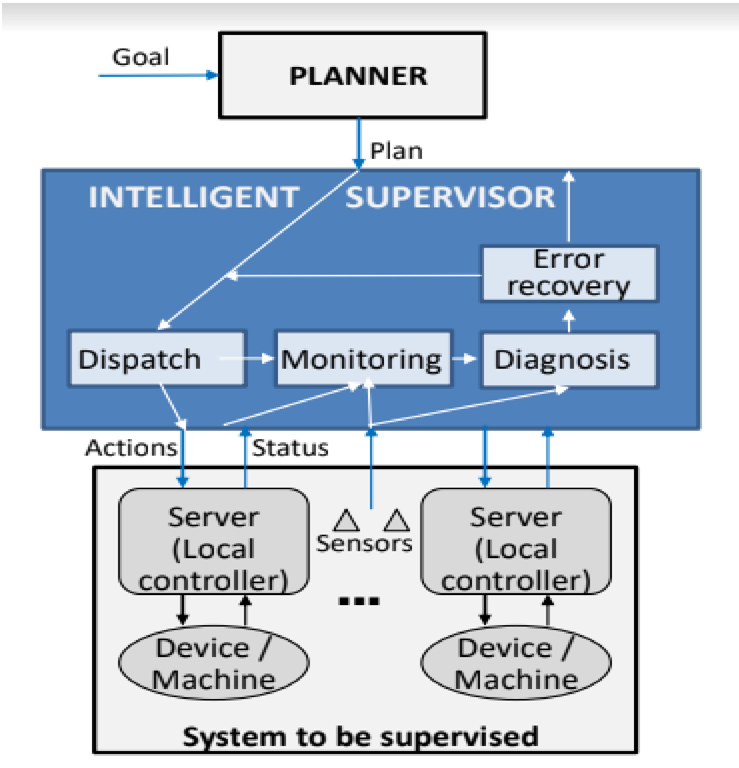


Figure 2.1 – Basic Architecture for an Intelligent Supervisor System by [40]

According to the architecture of the intelligent supervisor in Figure 2.1, it is possible to observe that it has at least four components that play a crucial role in monitoring, detecting and recovering errors: Dispatch, Monitoring, Diagnosis and Recovery Error. These components are described below.

The **Dispatch** is responsible for communicating with local controllers coordinating their actions. This component contains the updated representation of the status of the various systems and organizes the information for the next action to be performed [40].

The **Monitoring** has the function of detecting differences between the result obtained by an action and the expected result. This component receives information from the sensors to detect the current state and compares it with the state that was expected to be obtained after the action sent by the Dispatch.

Using the example above, after information has been sent to turn on the heater, the Monitoring reads the sensors to see if the room temperature has risen, if it has not risen, sends information to the Diagnosis.

After receiving the information, the **Diagnosis** tries to find out why the result of the action is not what was expected. To be able to diagnose, this component will have to receive additional information from the sensors and try to associate this information with a diagnosis, this may be possible with access to previous experiences or by applying machine learning [40].

In the example, the diagnosis can obtain information from the sensor that the window is open and diagnose that "the room temperature did not rise because the window is open".

The **Recovery Error**, after receiving the Diagnosis' diagnosis, tries to generate a recovery plan, if possible, or sounds an "alarm" to warn a human operator about the existence of the error. The self-healing system, in order to recover from errors, already must have a plan or a set of a-priori rules to solve that error [40].

A self-healing system is known as a system that is able to detect an error inside and, without external assistance, heal itself and go back to perform normally [41].

In the example, the solution would be to close the window, but if the windows are not automatic the system cannot close them. However, wasting energy with the heater on while the window is open is not efficient, so a possible recovery plan, in addition to the "alarm", would be to turn off the heater while the window is open.

In conclusion, with this chapter an overall introduction to the project concept and objectives is given, as well as some of the existing research and projects in the scope of the Ambiosensing project are presented.

The work explained in this document is the development of the application that interacts with the Intelligent Supervisor described above. The app has the propose to allow the users to see the data that the system is collecting, as well as give feedback that is used to improve the supervision system.

3. System Specification

In this chapter existent development methodologies for the app are presented and discussed including the one used to develop the application that interacts with the supervision system. The requirements for the app and UML diagrams are also presented in this chapter as they were also part of the first stages of development.

3.1 Development Methodologies

Software methodologies are seen as the way to utilize a group of methods to fulfill a goal [42]. The coordination between these methods originates a dynamic planning guide for the realization of the project, avoiding subjectivities in the process and increasing the productivity in the production of the project.

There are several models for software development. The model chosen for this project was the one that better fits its development conditions.

3.1.1 Waterfall Model

Waterfall model, illustrated in Figure 3.1 it is considered to be "the classic development model" and it's the oldest development process [43]. This methodology stands that one should not start a new phase without completing the previous one.

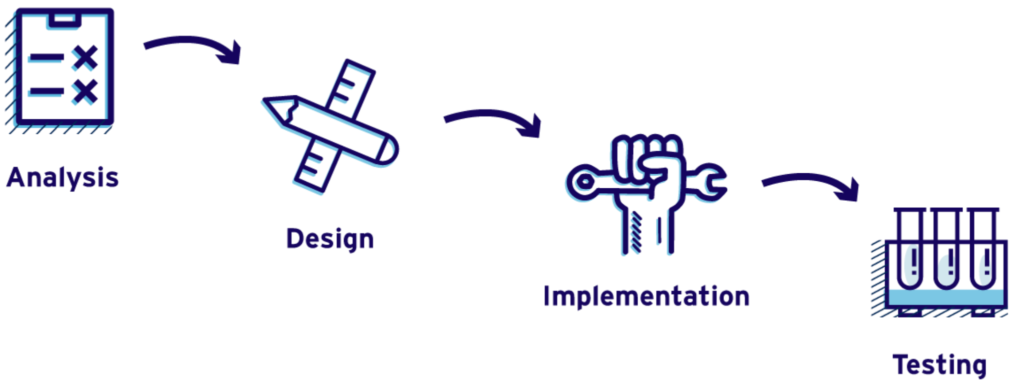


Figure 3.1 - Waterfall model [72]

The main advantages of this methodologies are:

→ The separate activities allow several programmers to have a part in the implementation of the project;

→ In case the specifications are well defined in the beginning, the development process can be done very efficiently.

Therefore, it can be assumed that this methodology is not the most suitable for this projects since it's common to add or change somethings along the way, or find errors at certain stages that, to be solve, need that something is altered in a previous step, and to go back and change something requires the entire process to be redone from the beginning [44].

3.1.2 Incremental Model

This model, presented in Figure 3.2, comes up as an update for the waterfall method. Contrary to the previous method, instead of developing the entire project in one go, the project is being developed gradually.

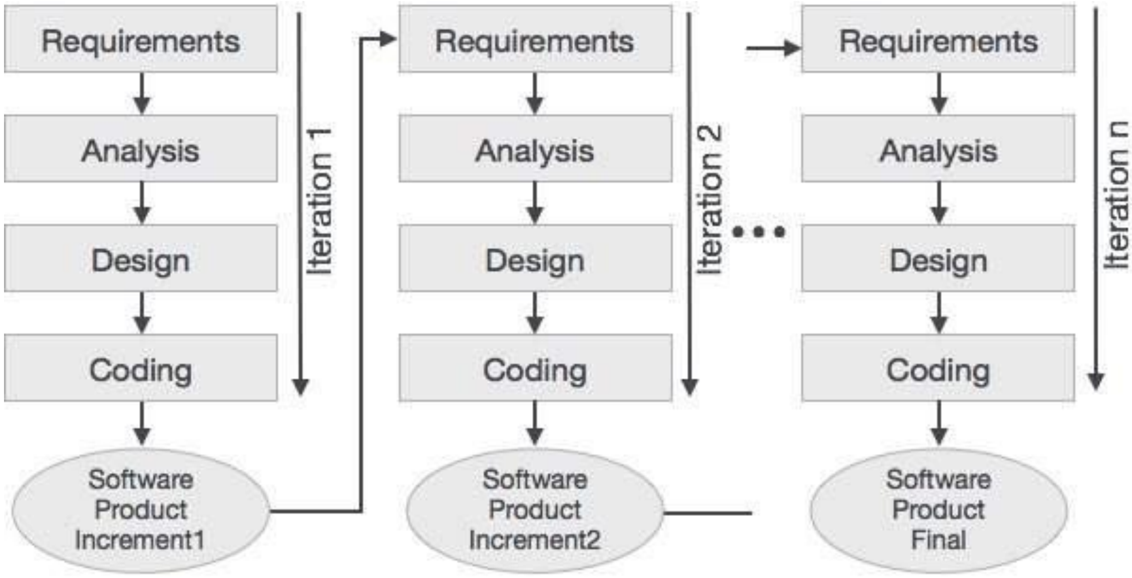


Figure 3.2 - Incremental model [45]

This model is divided into increments, each increment is developed in series (waterfall model) and delivered to the client for feedback. If it is necessary to change anything in the implementation, another increment is developed from the previous one, considering the requested changes. This process repeats itself until the project is concluded [46].

The main advantages of this methodology are:

→ The client has access to the system before it is totally completed; and

→ It is easier to obtain the client's feedback and implement the necessary changes to the system.

A problem with this model, other than being too heavy for developing certain projects, is that its structure tends to degrade with the addition of new increments.

3.1.3 Rational Unified Process (RUP)

In model Figure 3.3 it is illustrated the RUP model with the main objective to have more control over the results and the management of changes.

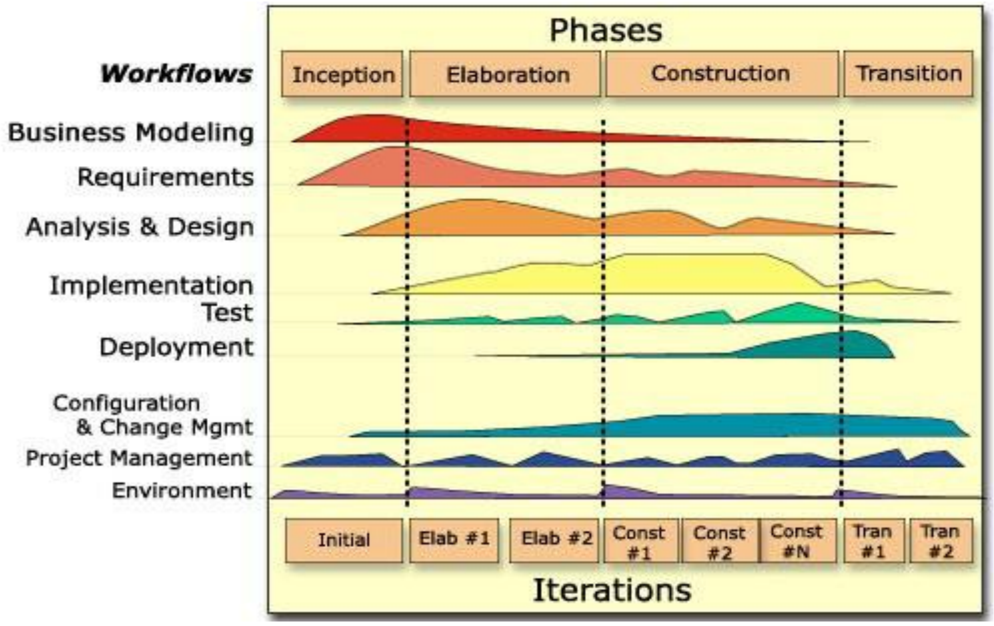


Figure 3.3 - Rational United Process [47]

The main feature of this model is that it is interactive and incremental.

An interactive project is a process that progresses through successive attempts and refinements. As such, this model is composed by various software development activities that are made along four phases: Inception, Elaboration, Construction and Transition. In these phases there can be various interactions.

Depending on which development phase you are you will have a higher focus on certain activities compared to others. For instance, in Inception there's a higher focus on Business Modeling and Requirements because the development process is in the beginning so the focus should be on what the project needs to present [48].

The Elaboration is the phase where the projection of the system is made, the documentation for the project is analyzed and revised, emphasizing the project architecture. This phase focus more in the activities of Analysis and Design of the application.

In the Construction phase starts the development of the software, programming and in the last interactions some tests are made (more emphasis in the Implementation and Test activities).

The last phase is when the software is delivered, and a plan of Deployment is performed.

For the development of this application the closest methodology used was the RUP. From the beginning, the testing process occurred alongside the implementation so that any bug could be fixed right away. The list of requirements was also not set from the beginning and so the documentation and codding were updated to accommodate the new requirements.

3.2 Functional and Non-Functional Requirements

As explained in the previous chapter, according to the methodology chosen, one of the first steps in developing an application is to get the requirements.

In the development of a project, it is necessary to have a set of activities composed of methods and procedures, in order to carry out projects that meet the requirements of customers.

The definition of requirements, according to IEEE Standard Glossary of Software Engineering, [49] is:

(1) A condition or capability needed by a user to solve a problem or achieve an objective.

(2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.

(3) A documented representation of a condition or capability as in (1) or (2).

In other words, the requirements of a project are the client's request of what the software must do to meet the client's needs.

This is the base phase from which the rest of the project is based, the requirements must therefore be well understood and documented so that there are no errors or delays in the next development stages. Requirements are divided into two categories, **Functional Requirements** and **Non-Functional Requirements**.

Functional requirements are, basically what the system must do to attend the costumer's requests [50]. They are features that allow the system do run as it is supposed to.

The functional requirements for this project include the features the app allows the user to do (Table 3-1) and the other functional features that must happen without the user's direct interaction (Table 3-2)

Table 3-1 - User functional requirements

Requirement	Explanation
Register	The user must be able to register in the application
Log in	The user must be able to log into the application with its account
Choose a space	The user must be able to choose a space, between the ones available with registered data.
Give feedback	The user must be able to give its feedback, about the space conditions.

Table 3-2 - System functional requirements

Requirement	Explanation
Display available spaces	Display the building's available spaces that have registered data value.
Display data values	Display the most recent data values registered in the selected space.
Display feedback's mean value	Display feedbacks mean value in the selected space.
Display graphs	Display graphs with the values registered during that day in the space.

The supervision system is monitoring various areas of a certain building and stores the environmental data in a database. There are various types of environmental data that can be stored in the database, Temperature, Humidity, Luminosity, Air Quality, CO2 levels and Energy Consumption, and their values should be accessed and displayed by the software for the user to see.

If Functional requirements are "what" the application must do the Non-Functional are "how" the application must do it [51].

Non-Functional requirements refer to the technical requests for the software. Security features, performance, error prevention, etc.

The non-functional requirements for this project are described in Table 3-3.

Table 3-3 - Non-functional requirements

Requirement	Explanation
Database connection	The app must be able to communicate with the database to extract data
Dynamic layout	The app must be able to adapt to the number and type of available data in every space.
Able to save user's feedback	The app must guarantee the user's feedback about every space is saved in the database.

The available types of environmental data can change in different areas of the building, there for the number of different values displayed can change so the layout must change accordingly (dynamic layout).

The user’s feedback saved in the database can then be used by the supervision system to adjust any parameters necessary.

The application is also only required to run in Android Operating System, and this condition will be very important and taken into consideration when choosing the development tool.

3.3 Conceptual Architecture

With the application requirements specified we can have an idea about the overall concepts that form the app and some of the connections that should exist between these concepts.

The overall idea is to have the application running on tablets. These tablets could be in the hand of the CEO or manager of the building, in the hand of an employee or on the wall for the costumers of the building to use.

As such, the application will have three types of users. One with the most control of what is shown and has access to the environmental data of the entire building/s, is called the Admin User. The second, Building User, would be the employee that has access to the environmental data of the entire building but cannot change what the other users see in the application. The last one, Space User, is for the applications that are running in tablets on the wall. These users will only have access to the environmental data of the space the tablet is in.

Having this general understanding of how the application should work for the different users a conceptual architecture was built, Figure 3.4, and is explained in the following paragraphs.

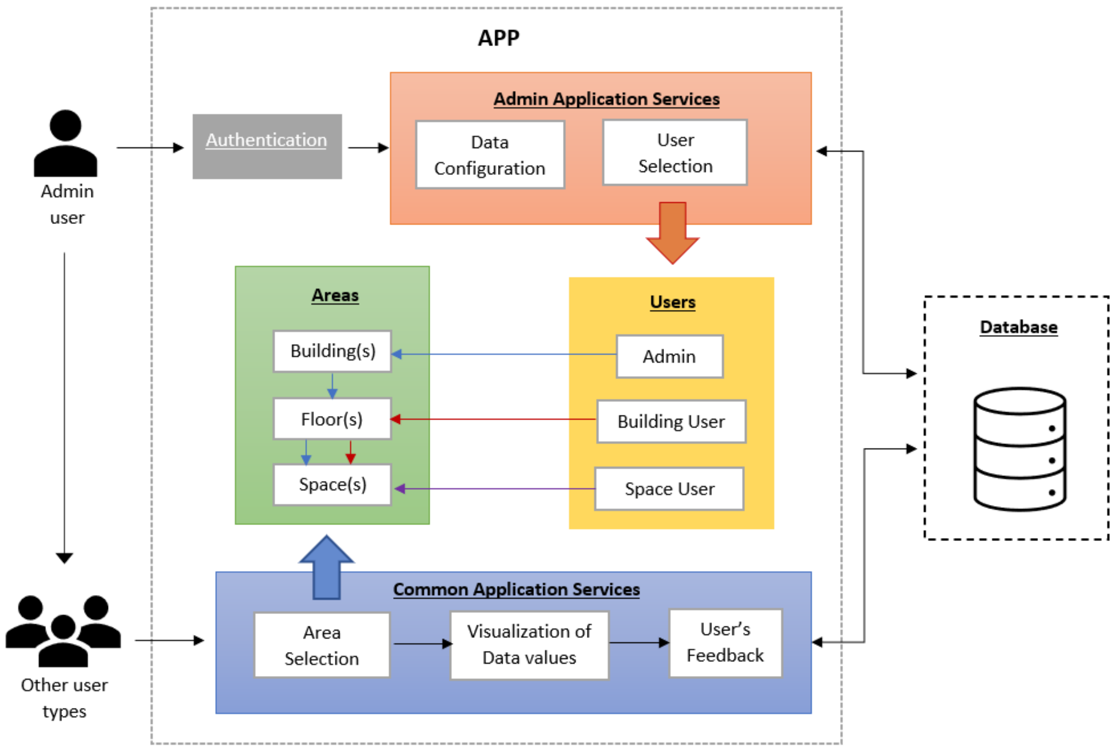


Figure 3.4 - Conceptual Architecture

The Admin user, must authenticate his entrance data before entering the app. There is then a set of services that only the Admin has access to like "Data Configuration" and "User Selection" to change the type of user.

By choosing the "User Selection" screen the Admin is able to select another user to operate in that tablet.

The user types that exist in this app are the Admin, with access to everything, the Building User that can only access a specific building, and the Space User that can only access a specific space in a building.

Except for the Space user that can only access a set space, all other users can select the area they wish to see data from, the Admin can pick any area and the Building User can pick any space in any floor inside its assign building.

All the users can visualize the environmental data of the spaces they have access to and give their feedback about the environmental conditions in those spaces.

3.4 UML Diagrams

Unified Modeling Language (UML) is a visual modeling language used for modeling application structure and software behavior [52].

UML was created due to problems resolving around software development and documentation. Even though it’s mainly used by software engineering, it’s now also been used to document business processes and workflows [53].

This language as two main categories: Structure Diagrams and Behavioral Diagrams [54]:

- **Structure diagrams**, such as the class diagram that will be presented, show the structure of the system showing its different objects and how they are related to one another.
- **Behavioral diagrams**, as the use case diagram and sequence diagram, show how the system behaves (or should behave) describing how the objects interact with each other to achieve the system’s goal.

3.4.1 Class Diagram

Class UML diagram is a common diagram used to document every software based on object-oriented programing. This diagram contains classes, each class has its attributes and operations, the different relationships between classes are shown with different types of arrows [54].

In Figure 3.5, there is the class diagram made to module the structure of this application.

The class “Building” it is where some information about the buildings that have a supervision system is stored, the information used by the application is the buildings id, to easily identify the building, and its name, to display to the user. A building can have one or many floors, and that is the relation between this class and the “Floor” class.

Both the classes "Floor" and "Space" have the same identifying variables as the building class and have a one to one or many relationship, which means that a floor can have one or many spaces associated to it.

The “Space” class is associated with the "Feedback" class and with the "Data" class, each with a one to zero or many relationship, which means that associated to a space can be 0 or many registered data values and 0 or many feedbacks from users.

In the "Feedback" class you only need the feedback values registered of every space.

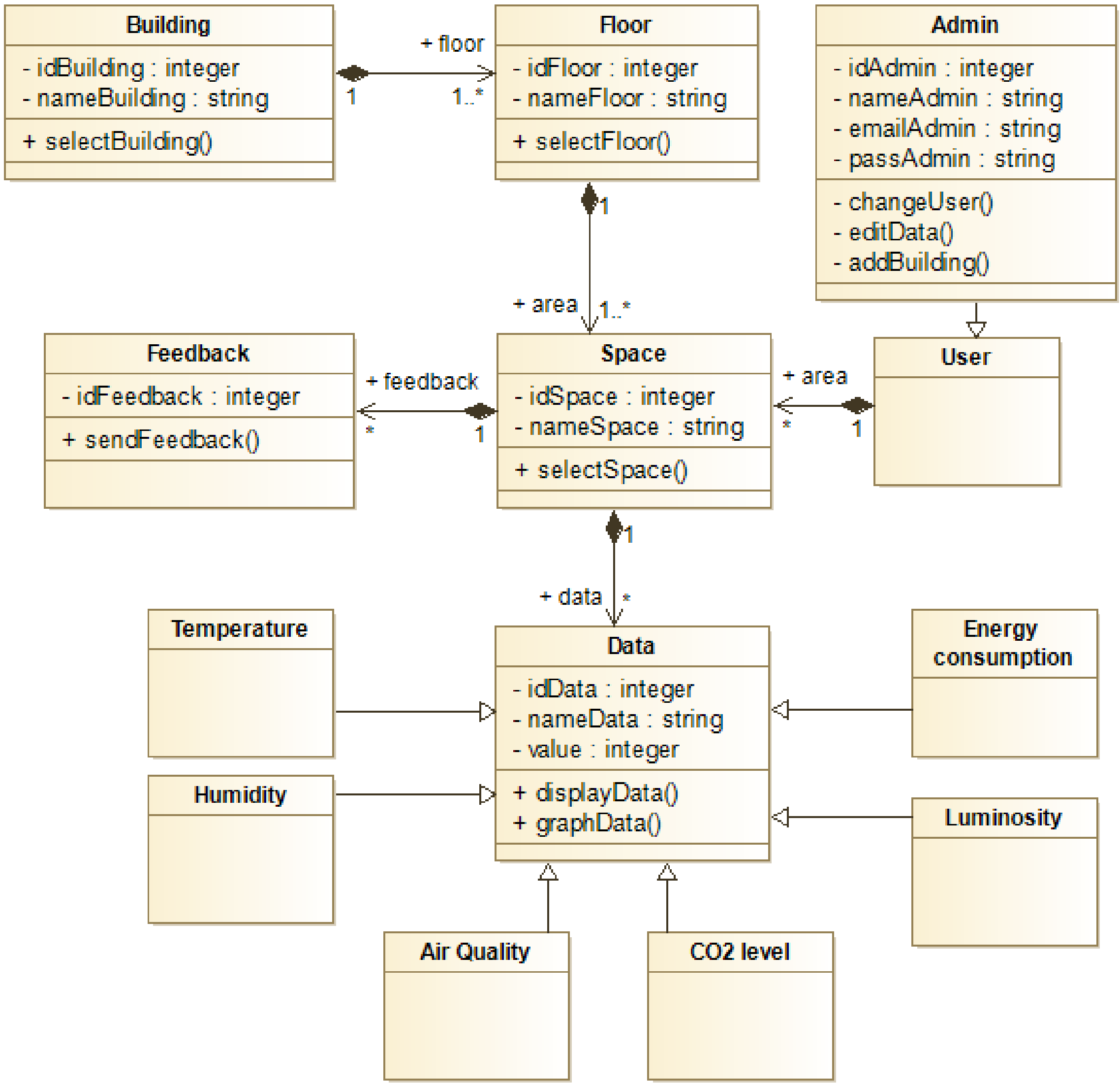


Figure 3.5 - Class Diagram

The "Data" associated with a space can be "Temperature", "Humidity", "Air Quality", "CO2 levels", "Luminosity" and/or "Energy Consumption". Any of these has an id for identification, a name, and a value to be visualized in the app.

The class "Admin" represents the user that can make changes as to what the other users can see or not in each space of the building. The "Admin" is a type of "User" that is related to the "Space" with a zero to many relationship.

3.4.2 Use Case Diagram

A Use case diagram is the most known behavioral UML diagram, it gives a graphic overview over the agents that interact with the system and the different functions needed for the system to work [44] and [52].

In this diagram it is easy to identify the actors and the main process of the system making the general idea of the program clearer from the beginning.

In the diagram from Figure 3.6 three main Actors are set to interact with the application each representing a type of possible users for the application.

The Admin is able to log in into its account, the password is verified within the system and, if not correct, an error message is displayed. After the log in, the Admin can see the environmental data available in every space of its assign building(s), can change the data the other users can see in a space and can change the type of user that will be using the application to see the environmental data.

The Building User has access to the entire building being able to consult the environmental data of every space as long as it's available not only in the database but on the Admin constraints as well.

The Space User is restricted by the Admin to a space in the building, it can see the environmental data of that space if it is available.

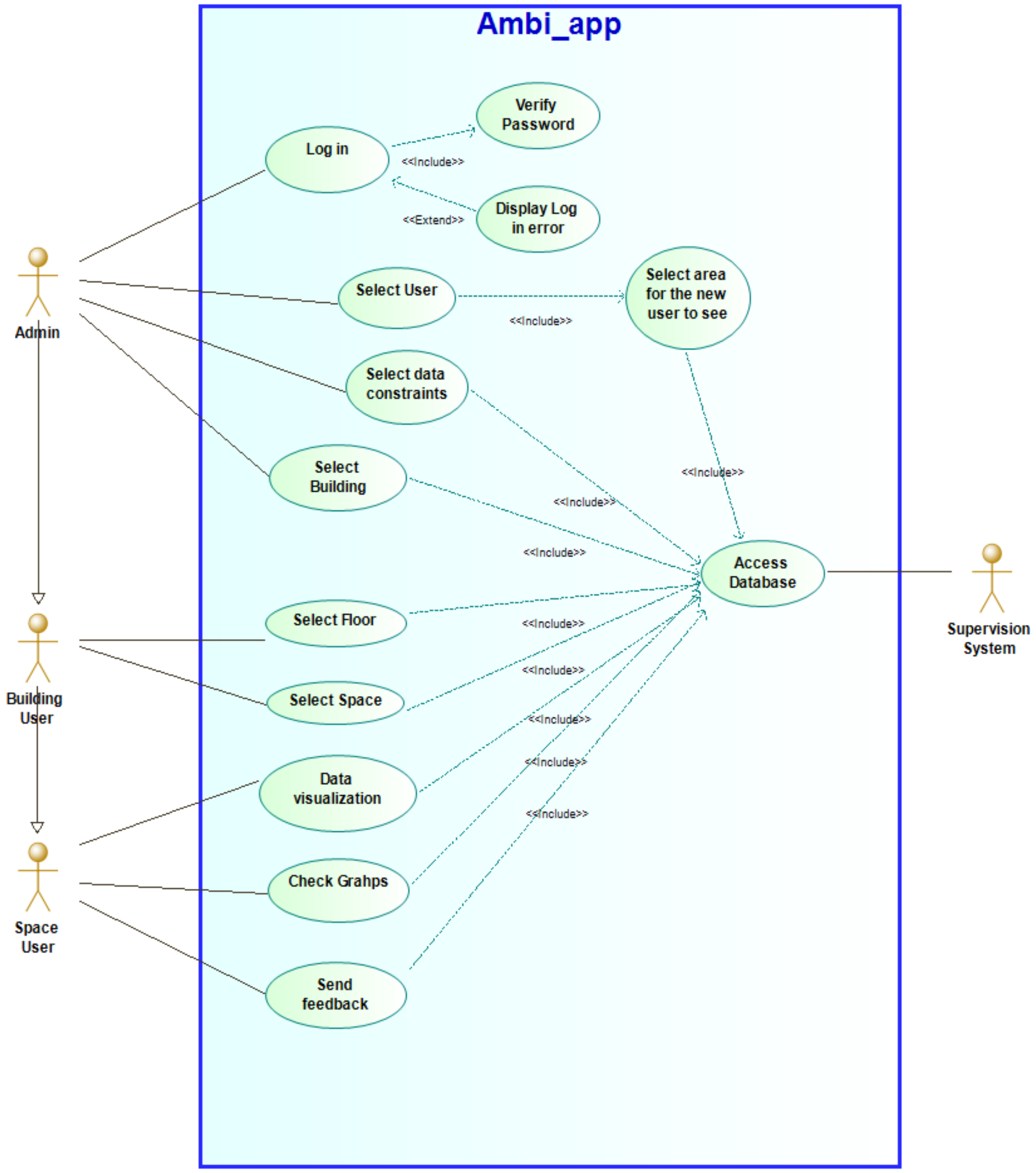


Figure 3.6 - Use Case Diagram

All users have access to the graph information and are able to give their feedback about an area.

Every building, floor, space and environmental data information are kept in a database that is connected to the secondary Actor, the Supervision System.

3.5 System Architecture

In this section, the general architecture for the developed app and system is described. Also, the flow of the process, of the developed app and system, on how the users of the app can use it to ensure that the space they are in has the ideal environmental conditions and give their feedback back to the system, is described.

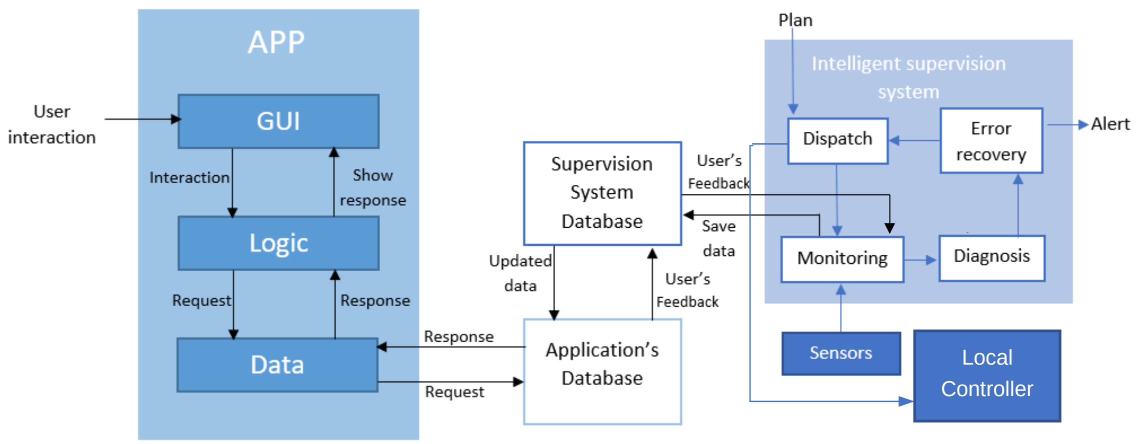


Figure 3.7 - App and System Architecture

In Figure 3.7 it is illustrated the connection between the system that is being developed in the Ambiosensing project, and the app that is explained in this document.

The user interacts with app by telling which areas he wants to see environmental data from. This interaction makes de Logic Tier request to the data tier the necessary information to display, be it the information about the building the user picked or the environmental in the space the user chose to see.

First, the user picks one of the available buildings, then the floor and space. The information about the buildings and spaces inside the buildings are kept in the app 's database (Figure 3.8).

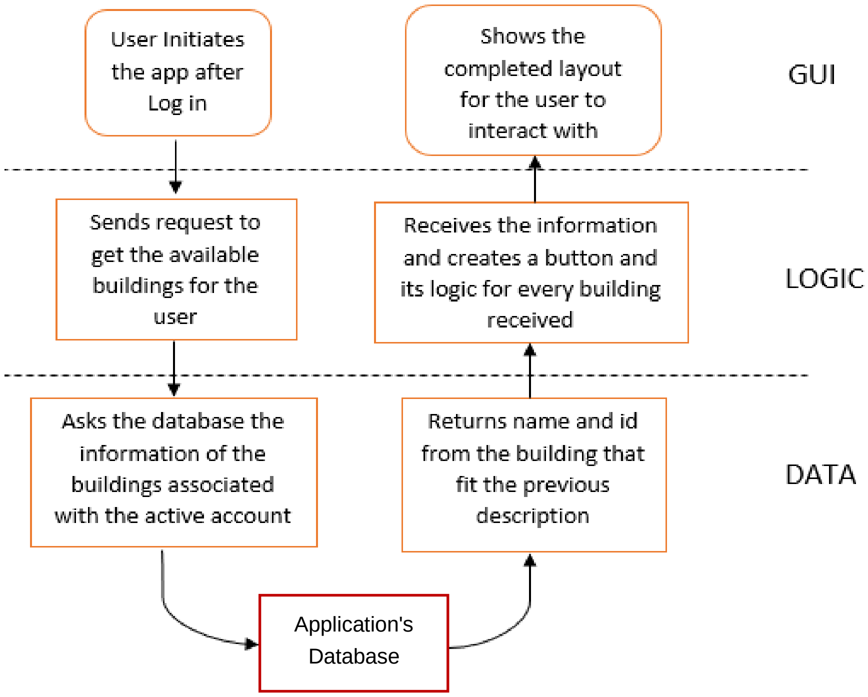


Figure 3.8 - Schematic for getting buildings information

Only the tables with environmental values and descriptions are updated with the systems database information (Figure 3.9).

The environmental variables are read by the sensors which sends the information to the Monitoring component from the supervision system. This component, additionally to the functions described in the chapter 2.5, will also save the information it gets from the sensors in the database (Figure 3.7).

This database from the system will help feed the app's database so that It can be accessed by It and shown.

The user can see this data by login into the app and choosing the specific area with those sensors.

Other than only seeing the environmental data, users are also able to give their feedback about how they feel in the space, how much is their level of satisfaction relating to the environmental comfort.

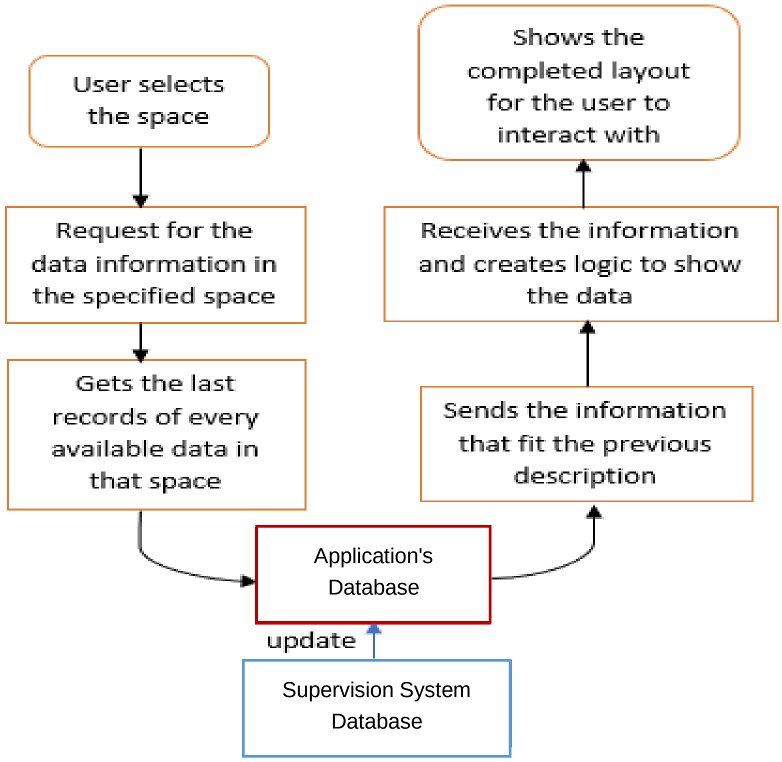


Figure 3.9 - Schematic for getting environmental data information.

The user can choose between five different levels of satisfaction, being one very low and five very high. This feedback is saved in the app's database and used to improve the Intelligent systems performance. The mean of the collected feedback for a space is also displayed in the layout that shows the collected data values.

4. System Implementation

4.1 Study of Technology

To develop the application that will have access to the data gathered by the supervision system, the most suitable technologies to develop and meet the required specifications for this project were chosen after comparing the available options.

4.1.1 Development Tool

There are several tools required to develop and test software, these tools include code editor, code libraries, compiler, and debuggers. To facilitate the development process and not use all these tools separately, developers usually use a single interface that consolidates all these tools into one software, these applications are called Integrated Development Environment (IDE). [55]

The benefit of using an IDE is to improve the developer's productivity by reducing the setup time and increasing the speed of development.

There is a big variety of IDEs that the developer can choose according to what the project is or what the coding language might be. There are IDEs for one specific language, IDEs to customize and develop mobile apps, cloud-based IDEs, etc. [56]

For this dissertation work, a mobile application is to be developed so an IDE for mobile apps is the most suitable. On this category we could choose to develop the app on a Native application or on a Cross-platform application [57].

A Native application development means to develop an app for one specific platform (Android, iOS, or Windows). The application provides specific UI design for the platform, allows to benefit from all available features of the platform and the application generally runs with high performance. The disadvantages are that it has a long development time as the code has to be adapted for every platform the app should be available on [58], [59].

A Cross-platform application will aloud the developer to share the same code with different platforms (Android, iOS and Windows). If this project app was to be available for multiple platforms this type of IDEs would have the advantage of being time saver and easier to deploy, because by utilizing the same code there is no need to adapt it to every platform. However, because they are not native, cross-platform applications don't integrate flawlessly with the operating system originating glitches in the performance [58], [59].

For the development of this project's app the IDEs Android Studio, IntelliJ IDEA, NetBeans IDE and Visual Studio – Xamarin were compared.

Android Studio is an integrated Android application development environment, is officially supported by Google and as such extremely easy to integrate Google Services (Amar InfoTech, 2018).

It only allows the development of android applications but in this case, it is not a problem as it is only for this operating system that we plan to develop the application.

Intellij IDEA is an integrated development environment developed by JetBrains, designed to enhance the programmer's productivity. It is extremely fast and includes several development tools. However, it is not free, and the open-source version has no support for JavaScript and HTML (Stone, 2018).

NetBeans is an open source, easy to learn, Java based IDE that allows developers to create desktop, mobile and web applications. It supports JavaScript, Php, C++, etc [60]. On the other hand, it is known to have a very slow debugger and has a lot if plugins that probably won't be needed but can't be uninstalled that add a lot of weight to the application [61].

Visual Studio is the integrated development environment from Microsoft and is available for free. It supports numerous programming languages, and it can be used to develop applications for Android, IOS and Windows, when combined with Xamarin.

Even though it is part of a growing community, the Xamarin community is smaller than the community of native IOS and Android development environments. In addition, there will always be a delay in supporting the latest platform updates (Altexsoft, 2019).

Table 4-1- Comparison of tools available for developing android applications

	Performance	Support for the Updates	Community	User friendly	Free	Type of platform
Intellij IDEA	🙂	🙂	🙂	☹	☹	Cross
Visual Studio - Xamarin	🙂	😐	😐	🙂	🙂	Cross
NetBeans	😐	😐	🙂	🙂	🙂	Cross
Android Studio	🙂	🙂	🙂	🙂	🙂	Native

Comparing the different IDE options, we can see that Intellij IDEA it´s a really good software but it's a bit expensive and can be complex to learn for people who are not experience in developing apps. For these reasons, it is out of our possibilities list.

Visual Studio – Xamarin and NetBeans are cross-platform IDEs and therefore as explained before the apps produced by them are more prone to have performance glitches and there is no need to take that extra risk if the app of this project is only for Android.

Android Studio, being a native platform IDE, provides the fastest tools for Android apps development, has the Android specific UI design so the user gets the expected looks of the platform and it has better performance. [59]

The application used for this thesis work development was Android Studio due to the reasons listed above.

4.1.2 Database

As previously described, this app will display the data collected by the supervision system. Nevertheless, as this system has its own tables, to save data it collects and extracts information, the database chosen for the app is the same as the system, MySQL. This way, this app entities are in the same database management system (DBMS) as all the other entities of the other Ambiosensing systems.

MySQL is an open-source relational database management system, it organizes the data into entities and these data types can be related to each other structuring the data, just like an Entity and Relationship Diagram.

Entity and Relationship diagram (ERD) was built by engineers so that they could have a visual way to understand how all the objects in a database are related and they are working together (Lucidchart, 2017).

An entity is a component of data and a collection of these can have attributes that define them, by defining the entities and showing the relationship between them, an ERD demonstrates the structure of a database (Terminology, 2009).

In the diagram built for this project's App (Figure 4.1) there is an entity for the admin user information "AdminUser" with the user information including its id. There is also a "Building" table with the information of every building with this project's supervision system.

These two entities are connected through a many to many relationship, because an Admin can have access to a lot of buildings and a Building can have many admin users having access to it, this relationship creates a joining entity "AdminUser_has_Building" that will capture every instance of related data between the admin user and the building.

Then one Building can have one or many Floors and each Floor can have one or many Spaces. There are many types of spaces, these are stored in the SpaceTypology entity and every type can have zero or more Spaces associated to it.

Every Space has zero or many Data values connected to it and every data type (DataTypology) can have zero or many Data values being stored. A Space can also have zero or many feedback messages associated to it and can have zero or many data that has been configurated by the admin to show (DataToShow).

MySQL uses SQL (Structure Query Language) which is a language used to create, extract, and modify the data in the database (Pisa, 2012).

To connect the, still developing app, to the database XAMPP software was used. XAMPP is a localhost that can work on a desktop or laptop which is very convenient to test MySQL, PHP and Apache projects [62].

Apache is a remote server that receives the requests and uses HTTP servers and serve what has been requested [63].

PHP is a language mostly used to create web-based software applications, its code is executed on the server and its HTML code is displayed on the browser side [62].

XAMPP is a software package that contains the web server (Apache), the relational database management system (MariaDB, wish is a fork of the MySQL) and programing language (PHP). All these necessary to test this project.

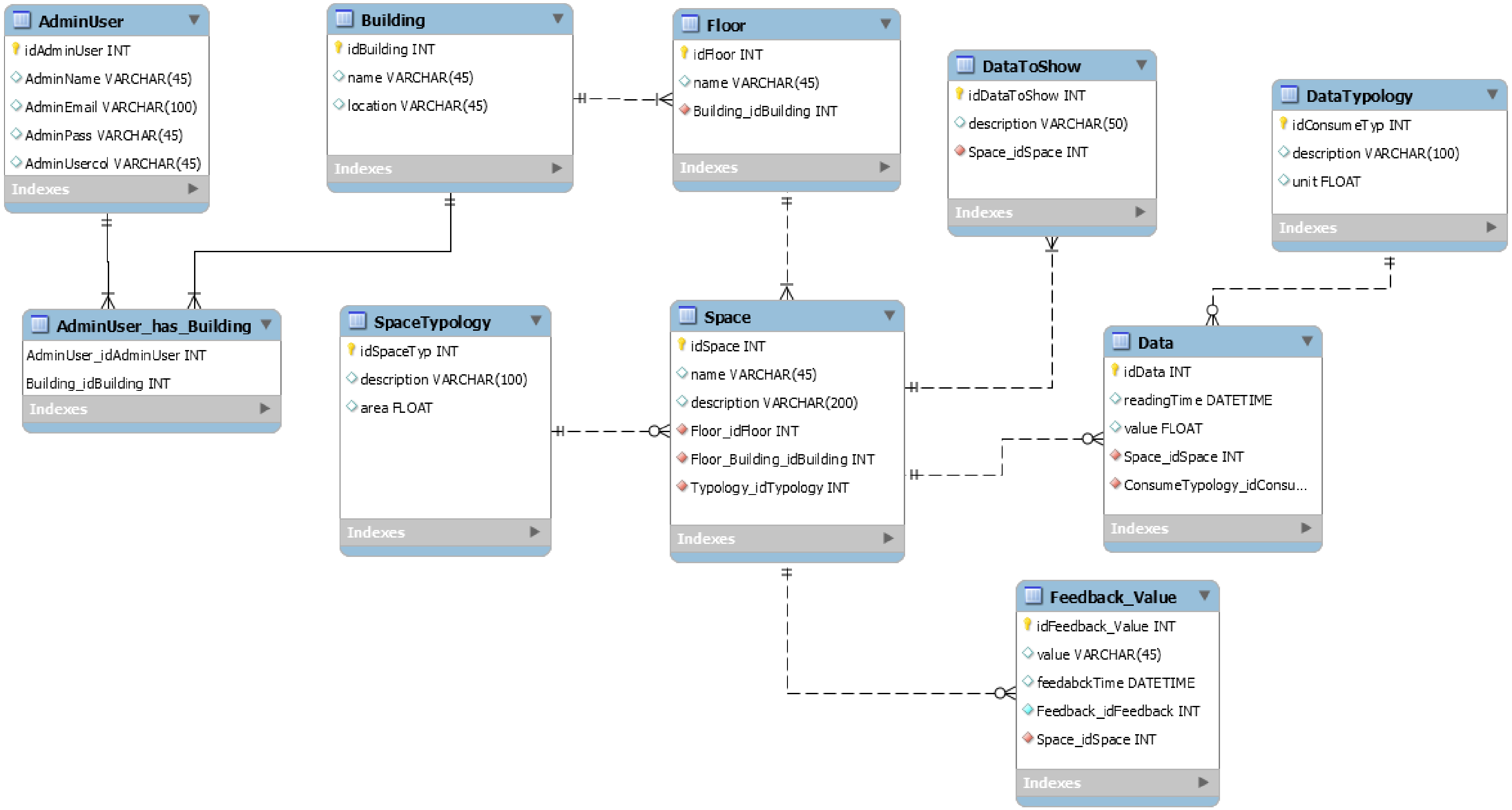


Figure 4.1 - Entity and Relationship Diagram

4.2 Development

The development phase is where we apply everything that was discussed in previous phases and put it on code. Reviewing all the requirements, structure and expected behavior for this app, a conceptual architecture is built with three different Tiers that will have to interact with each other to accomplish the final goal of this project.

In Figure 3.7, in chapter 3.5, the application was already divided in these three tiers, in Figure 4.2 we can see the type of files that compose each of these tiers.

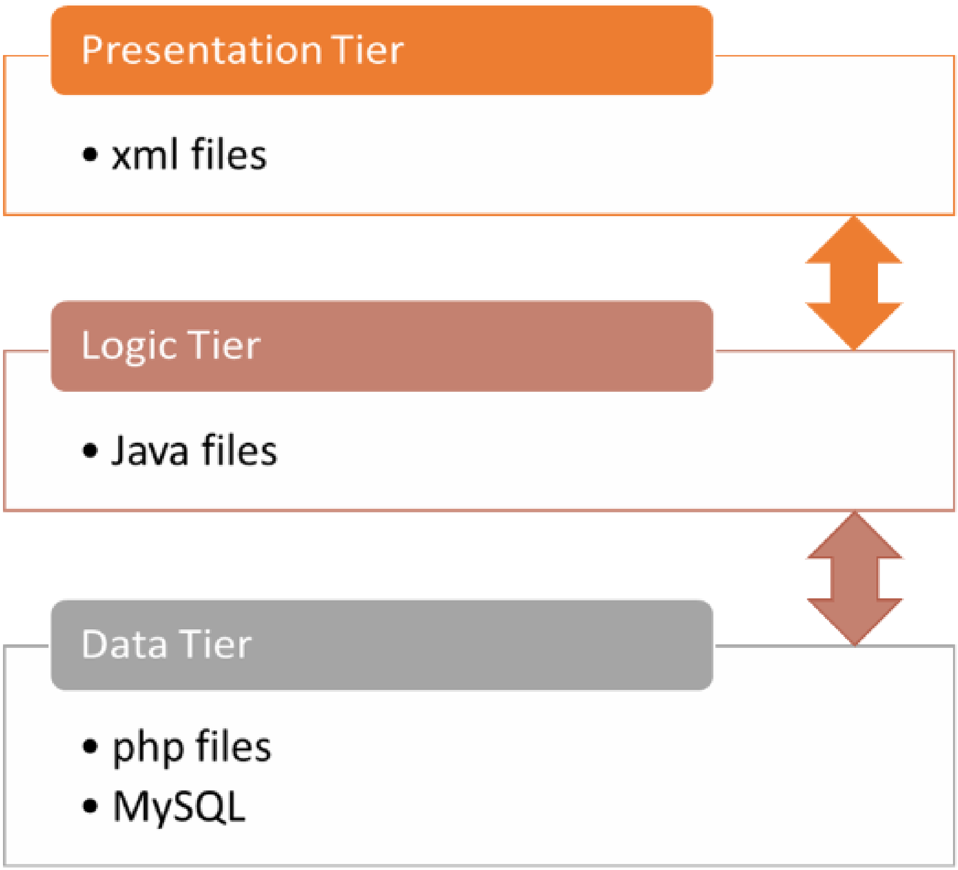


Figure 4.2 - 3 Tier App Architecture

The Presentation Tier it is the front-end layer in this 3-tier architecture that has the user interface. The graphic-user interface presented in this layer display the content and information with which the user will interact with. [64]

The Logic Tier contains the logic that conducts the application core abilities. This where all the conditions, associations, etc., are implemented to make the app work as it should.

The Data Tier contains the database/data storage system from which data is requested and obtained. This tier is also composed by the data access layer with which the logic tier communicates through API calls [65].

4.2.1 Graphical User Interface

Graphical user interface, as the name specifies is what the user sees and interacts with in the app, the presentation tier.

There were no specific requirements by the company about how the app should look so it was decided to go with a simple presentation design, to be easier to use by people.

The code for this tear is in xml but thanks to the drag and drop feature from the Android tool the code was almost all written by the software according to what would be added in, modified or removed from the display.

Although a big part of the design was made like this a concern that was raised early on was the need for a dynamic layout.

For instance, a space could have four different sensors, Temperature, Humidity, Luminosity and Air Quality, so it would show four different types of data (Figure 4.3).

Another space could have, other than these four types, a sensor to read the C02 and another to measure the Energy Consumption levels, so six different types of data to show.

Figure 4.3 - App's layout with four different sensors readings

These layouts, along with the layouts that would have the buttons to choose between the available buildings, floors and spaces (Figure 4.4), could not be static so they were made dynamically in java according to the information received from the database.

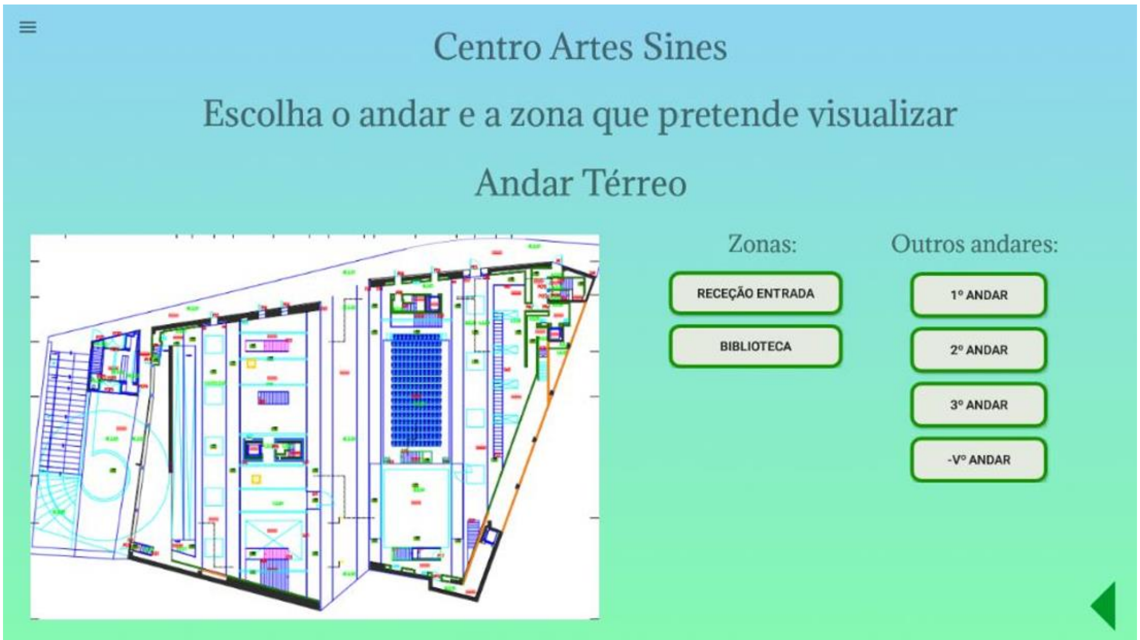


Figure 4.4 - App's layout to choose the floor and space (2 space options)

4.2.2 Logic Development

The logic part of the application was developed in JAVA in the chosen IDE, Android Studio.

Android apps are the combination of components that can be invoked individually, for example an Activity.

The Activity is a class that is invoked when the user interacts with the app and it is responsible for holding the user interface (UI) components. Generally, one Activity implements one screen, so apps with a lot of screens comprise multiple Activities [66].

In this project, however, only three Activities were implemented each having Fragments to control layouts. Fragments are representations of behaviors or portions of an Activity. One Activity can have multiple Fragments, build a multi-plane layout and reuse fragments in multiple activities [67].

According to [68], you should use Fragments if possible, since it makes the maintenance and control of the code much easier but in some cases, many Fragments to one Activity can get to complex as the Fragments have to communicate with the host Activity.

This app has the mainActivity, that is the first class to be invoked by the user, here there are eight Fragments associated that perform logic in eight different layouts.

Another Activity is the dataActivity, that has three fragments associated, these fragments are the ones that controls the environmental data to show and other two related to the feedback, where the user can send his/her feedback to the system.

The last one is the graphActivity, that has a multi-plane layout with a Fragment that shows the readings in a graph, showing different graph for different types of data (Figure 4.5).

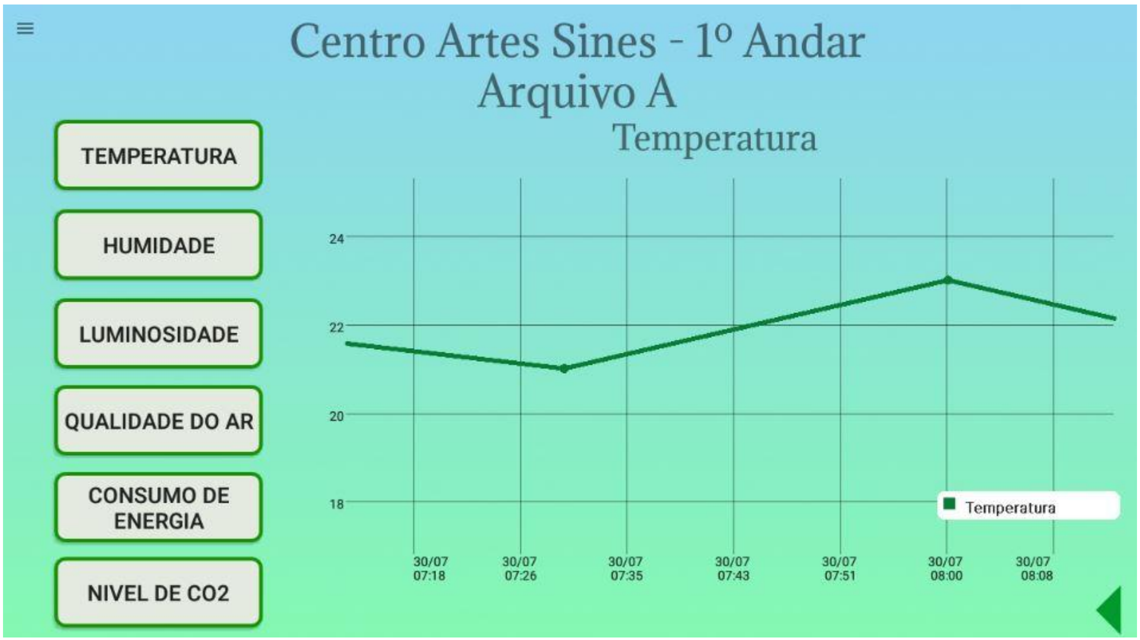


Figure 4.5 - Graphic Layout

The app starts in the mainActivity and tries to connect to the database checking if the IP that is stored is the right one. In case it is not, it will show the Fragment with the layout to ask the user for a new IP. Once this IP is checked the login layout appears (Figure 4.6), which is another Fragment and from here the user can go to the Fragment to change the IP again or go to the Fragment to register a new account.

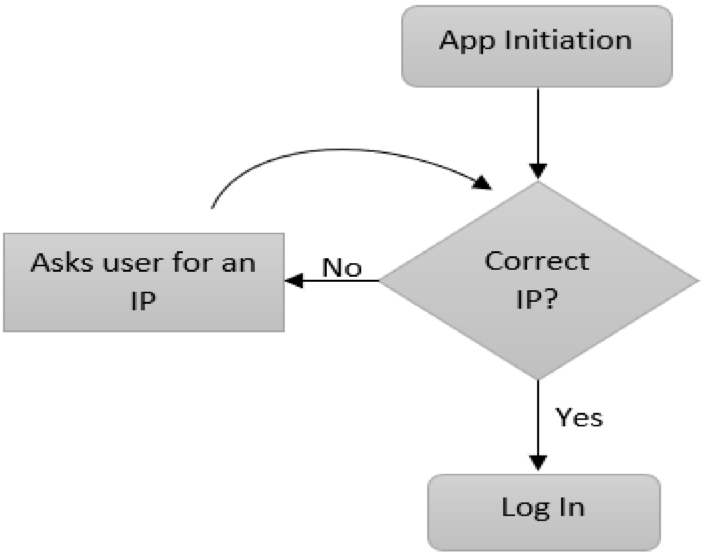


Figure 4.6 - IP fluxogram

Login data and IP are kept in a Shared preference object that points to a file that provides simple methods to help write and read saved key-values pairs. These are managed with the session management which stores the information outside the app allowing this data not to be lost when the user closes the application (Figure 4.7).

```
if(success.equals("1")){
    Toast.makeText(getContext(), text: "Sessão de Administrador iniciada!", Toast.LENGTH_SHORT).show();
    loading.setVisibility(View.GONE);
    sessionManager.setLogin(true);
    sessionManager.setUsername("admin");
    sessionManager.setIdAdmin(jsonObject.getInt( name: "idAdmin"));
```

Figure 4.7 – Save the Admin log in state with session manager

When logged in, the user can choose between the buildings associated to his account which one he wishes to see environmental data from, after choosing the id related to the building is sent to another Fragment.

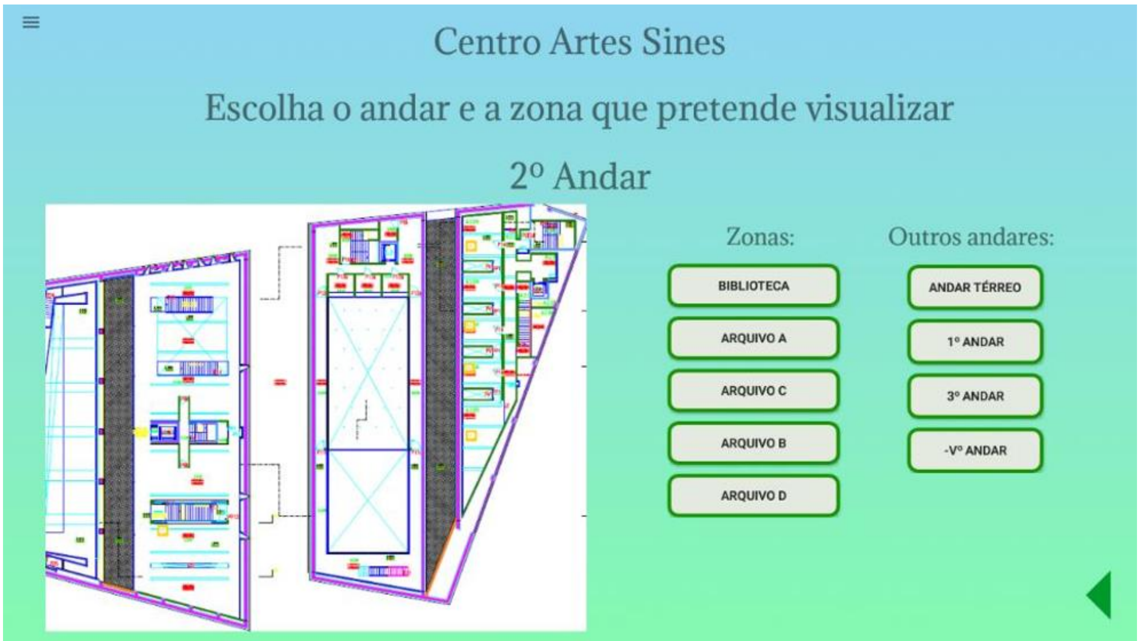


Figure 4.8 - Floor and Space layout

This shows another layout where the user is able to choose the floor and space being the options available the ones that are associated to the chosen building's id (Figure 4.8).

```java
FragmentTransaction transaction = getFragmentManager().beginTransaction();
floor_frag fragInfo = new floor_frag();
Bundle bundle = new Bundle();
try {
    bundle.putString("building", String.valueOf(products.getInt( name: "id")));
    bundle.putString("floor", String.valueOf(0));
    bundle.putString("name", products.getString( name: "name"));
} catch (JSONException e) {
    e.printStackTrace();
}
fragInfo.setArguments(bundle);
transaction.replace(R.id.fragment_container, fragInfo);
transaction.addToBackStack("buildings");
transaction.commit();
```

Figure 4.9 - Sending the chosen building's information to another Fragment

In Figure 4.9 and Figure 4.10 there is an example of code to send and receive information between two fragments.

```java
Bundle bundle = this.getArguments();
idBuilding = bundle.getString( key: "building");
name_build = bundle.getString( key: "name");
building_name.setText(name_build);
floor = bundle.getString( key: "floor");
```

Figure 4.10 - Receiving and saving the information received by the previous Fragment

These two layouts (choose building, floor/space), as said before are created in java according to the received building data from the database.

Every time the user selects, in these layouts, another floor, the Fragment replaces itself with the same Fragment/layout but with new initial information that allows the display of the available spaces in the new selected floor and also the display of other available floors in the building (Figure 4.11).

```
buttons[i] = new Button(getActivity());
buttons[i].setId(j+1);
LinearLayout.LayoutParams layoutParams = new LinearLayout.LayoutParams( width: 300, LinearLayout.LayoutParams.WRAP_CONTENT);
buttons[i].setBackgroundResource(R.drawable.button_border);
layoutParams.setMargins( left: 40, top: 10, right: 0, bottom: 0);

buttons[i].setLayoutParams(layoutParams);
if(products.getString( name: "name").equals("0")){
    buttons[i].setText("Andar Térreo");
}else
    buttons[i].setText(products.getString( name: "name")+"º Andar");
constraintLayout.addView(buttons[i]);
```

Figure 4.11 - Example of code to create the dynamic buttons

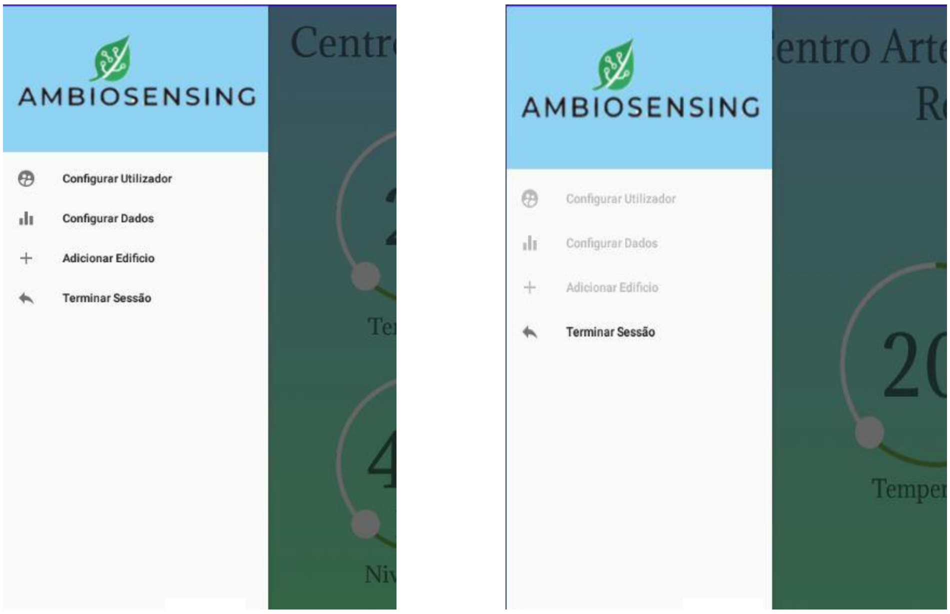
This app also has a Navigation Drawer, which can be seen has a menu, mainly for the admin to navigate through the Fragments that allow him to make changes in the app.

This menu is accessed from the top right corner of the screen and is controlled by an Activity, in this case three Activities.

If the user is the Admin, these options are unavailable for other types of users, he can use this menu to get to the different Fragments (Figure 4.12 [a]). One that allows to hide or unhide environmental data information in a selected space, another to input a building code giving this admin aces to that building and one fragment where the admin can configurate the tablet to be used by another user.

A building user can move freely throughout the chosen building by the admin and the space user can only see the environmental data from a specific place chosen by the admin. The information for the building id and the floor id and space id (in case it's the space user) is saved with the session manager so that, if the app closes and opens again, it will continue that user's configuration.

These other users can only use the menu to logout from the app, after that an admin has to login again to use the app (Figure 4.12 [b]).



[a] [b]

Figure 4.12 - Drawer Menu when the admin is logged in [a] and when it's another user [b]

The dataActivity initiates when the user chooses a space to see or when the Admin configures the tablet for the space user. In these both cases the activity receives the id of the building, the floor and the space and passes this information to the Fragment that will deal with the display of the data, this Fragment connects to the database to get not only the environmental data but also the mean feedback for that space´s environment.

Figure 4.13 - Data layout

By selecting the Feedback button (Figure 4.13), a fragment with the available options of feedback is initiated and detects when an option is chosen, saving the respective value (1 to 5) of feedback in the database (Figure 4.14).

Figure 4.14 - Feedback layout

This activity is set to update from 30 to 30 seconds guarantying that the data showed is the most recent one.

When the button "Gráficos" (Figure 4.13) is selected the *ids* information are passed to the graphsActivity.

The layout associated with this Activity is a bit different has it is a multi-plane layout. The buttons to choose the type of environmental data are controlled by the Activity class itself, while there is a portion of the layout, where the graph and its title are, that is controlled by a Fragment class (Figure 4.15).

The Fragment gets the environmental data from the database and displays the graphic and respective title. When a button to see another type of environmental data is selected, the Activity replaces the Fragment with the same Fragment but send the information of this different type that was selected and only this part of the layout changes.

Figure 4.15 - Graphic layout with fragment representation

All the communications this application makes with the database are made by scheduling network request with the VOLLEY library. The API requests are made connecting the app to the database through the php files (Figure 4.16).

```
StringRequest stringRequest = new StringRequest(Request.Method.POST,
        url: "http://"+ip+"/android_register_login/connectTest.php", new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        try {
            JSONObject jsonObject = new JSONObject(response);
            String success = jsonObject.getString( name: "success");
            if(success.equals("1")){...}
            else{...}
        } catch (JSONException e) {...}
    }
}, (error) → {
        Toast.makeText(getContext(), text: "Errol! "+error.toString(), Toast.LENGTH_SHORT).show();
}) ;
RequestQueue rqt = Volley.newRequestQueue(getContext());
rqt.add(stringRequest);
```

Figure 4.16 – Example of a Volley request to get information from the given URL

4.2.3 Database Connection

To get the data needed for the app to perform accordingly, a PHP program was developed. PHP is used to connect with the server and get or save data in the database by evaluating MySQL queries [69].

After the database with its tables is created the connection PHP file was made to connect to it. This file is then accessed by other PHP files to enable the connection with the database, avoiding having to repeat this code for every file.

```
$conn = mysqli_connect("localhost","root", "", "ambi_ss");

if (mysqli_connect_error()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
    die();
}
```

Figure 4.17 - Connection PHP file

After calling the connection the program initializes a msqli_query with that connection and a statement, the statement will indicate, basically, what we want from the database.

For example, in this log in program, the email and password inserted by the user are received, and we say we want to select the id and password from table useradmin (where the user's information is kept), where the email is equal to the one received. If it is able to find the email and the password matches, the success message is sent to the JAVA tier. Otherwise, an error message is sent.

```php
$email = $_POST['email'];
$password = $_POST['pass'];
require_once 'connect.php';
$sql = "SELECT id, password FROM useradmin WHERE email='$email'";
$response = mysqli_query($conn, $sql);
$result = array();
if(mysqli_num_rows($response) === 1){
    $row = mysqli_fetch_assoc($response);
    if(password_verify($password, $row['password'])){
        $result["success"] = "1";
        $result["idAdmin"] = $row['id'];
        $result["message"] = "success";
        echo json_encode($result);
        mysqli_close($conn);
    } else{
        $result["success"] = "0";
        $result["message"] = "error";
        echo json_encode($result);
        mysqli_close($conn);
```

Figure 4.18 - Log In PHP code

All the other PHP files follow the same structure, only the sql statement, entrance variables and the output would change.

To get the data to display in the app the PHP receives the id of the space and the sql string is as Figure 4.19 demonstrates. What is needed to collect from here is the last inserted reading of every type in that space. Information from two entities was needed, data entity and data type entity.

The INNER JOIN keyword was used to accomplish this goal, this allow us to select records that match between these two tables [70]. The value from the entity data, the unit and description from the entity datatype were selected from the records where the keys binding these entities are the same.

In these records the ones that have the space's id equal to the entry variable can be selected. Another part to add is time, the most recent reading of every type available is needed.

GROUP BY is used to group rows with the same values into summary rows [71]. With this the maximum reading time in the environmental data entity where the space's id is the received one was acquired for every data type available. Enabling the extraction of the necessary records.

```
$sql="SELECT data.value, data.idDataTyp, data.idSpace,
datatypology.description, datatypology.unit FROM data
INNER JOIN datatypology ON datatypology.idDataTyp=data.idDataTyp
WHERE data.idSpace = $zone AND data.readingTime IN
(SELECT max(data.readingTime) FROM data WHERE data.idSpace = $zone
GROUP BY data.idDataTyp)";
```

Figure 4.19 - Get data PHP code

The dataToShow entity is where the environmental data the admin made changes on and he wants to display is saved. The entrance variables are the space's id where the change occurred and the environmental data descriptions that are supposed to be shown.

Before saving new records about a space, previous records of that space must be deleted. After that, the new records are saved in the database using the INSERT INTO statement.

```
$sql="DELETE FROM datatoshow WHERE idSpace='$idSpace'";

if (mysqli_multi_query($conn, $sql)) {

    foreach($data as $option) {
        $sql ="INSERT INTO datatoshow(description, idSpace) VALUES ('$option', '$idSpace')";
        $result = array();
```

Figure 4.20 - Data to show PHP code

This tier was much more intuitive to build compare to the other ones, because I already had worked with php and MySQL queries during the Electrotechnical Engineering course.

After explaining the development process the sequence diagram in the next chapter will help in the understanding of the app's flow.

4.3 Sequence Diagram

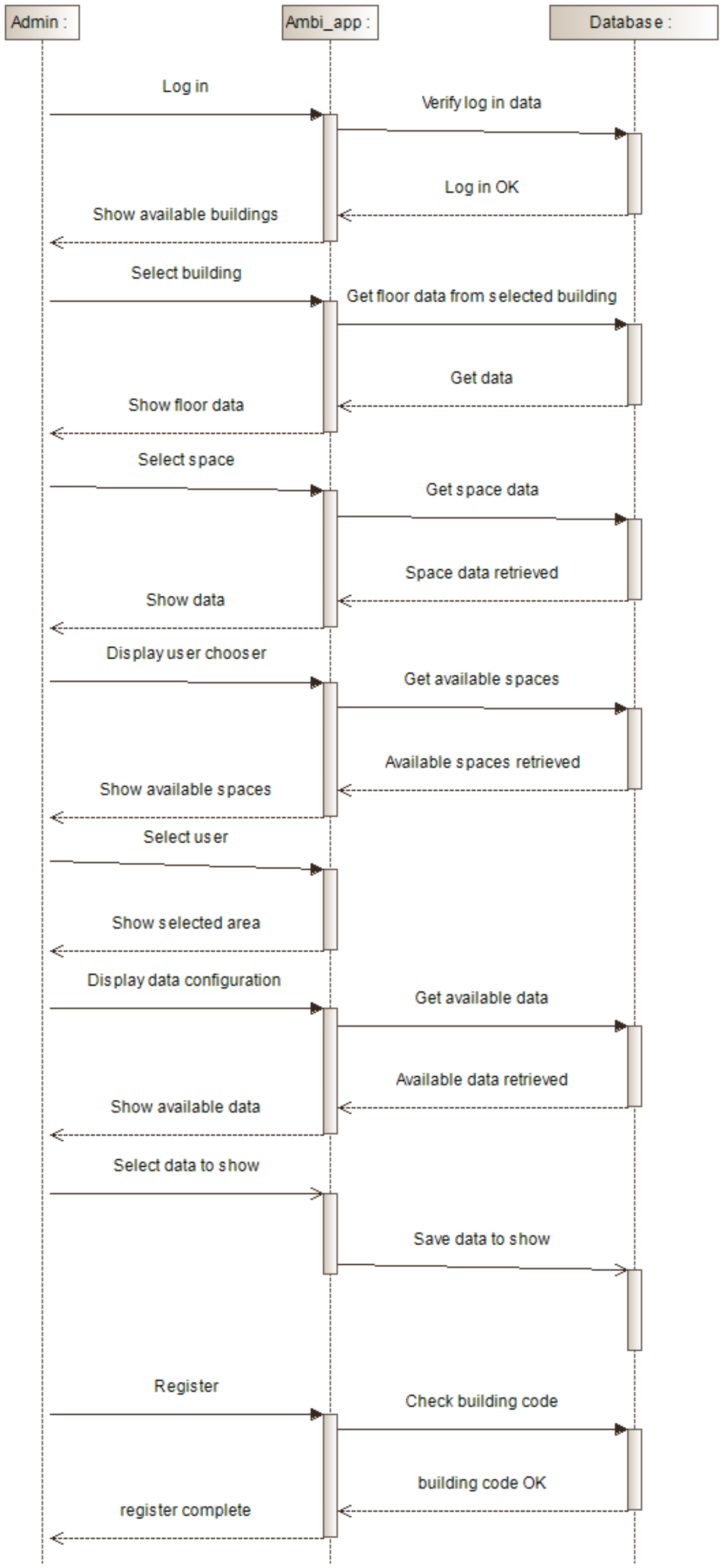


Figure 4.21 - Admin Sequence Diagram

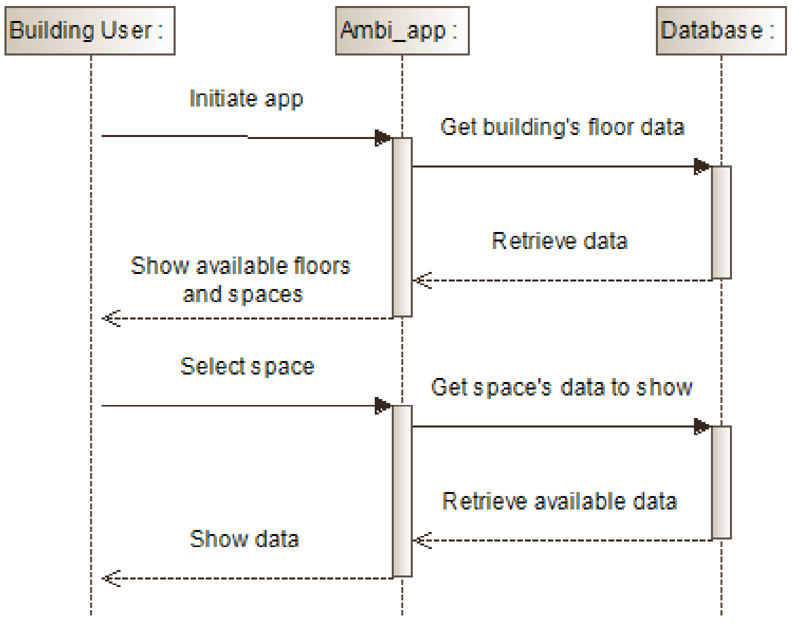


Figure 4.22 - Building User Sequence Diagram

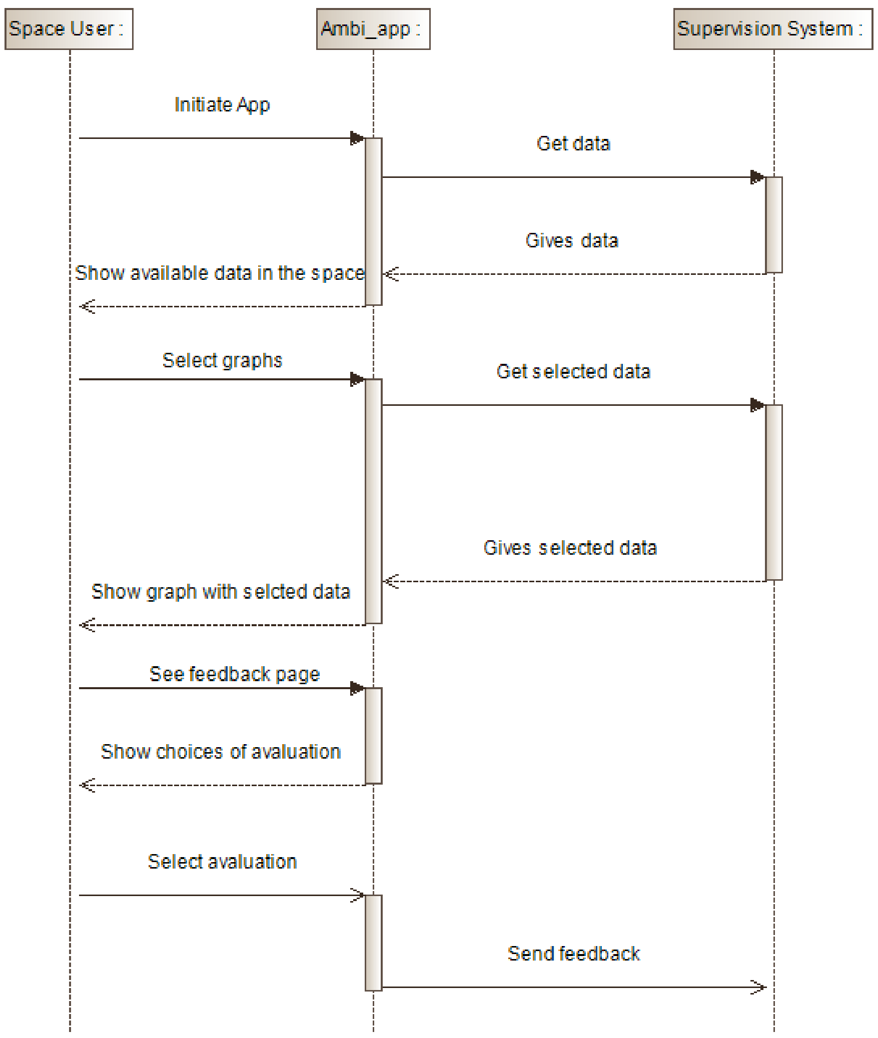


Figure 4.23 - Spaca User Sequence Diagram

Sequence diagrams are visually self-explanatory and because of that these models have become very popular not only for the computer science community but also in business application development [54].

This diagram shows how the objects interact with each other and in which order these interactions occur. Objects can be activated or deactivated depending on if other objects want to communicate with them or not.

For this project three sequence diagrams were made to represent the interactions of the three different types of user with the system (Figure 4.21, Figure 4.22, and Figure 4.23).

As explained in the Use Case diagram the Admin is the user that can interact more with the application (Figure 4.21). To Log in, the Admin requests the app to get in into its account. The app then requests the database for log in data verification. After validating, the database responds to the request by sending a "success" message. The app opens the Admin account and shows the available buildings for that account.

To select a building and a space, the Admin must make the request to the app, that then sends the request to the database to get the necessary data to show.

For the user selection, the Admin has to assign which space (for the space user) or building (for the building user) it wants the user to have access to and after selecting it, a confirm is send to the app that will change the user accordingly.

To configurate the environmental data available for visualization, after getting the available data in the database, the Admin can choose which to "hide" and which to show, after confirming, the app sends the updated list of available environmental data in that space.

After the Admin user configuration, if the selected user is the Building user a request is send to the app to show the available spaces in the building the Admin chose. The app requests the building data to the database and the database responds with the requested data that is then shown to the user by the app responding to its request (Figure 4.22). The user can then, chose a space and the app will request that space environmental data to the database.

If the Admin choses the Space user, the app will request the selected space available environmental data to the database that will then respond with the requested data and the app will respond to the user by showing that data (Figure 4.23).

The user can also request to see the graphic data available in the space, the app will request the data to the database to then show it in a graph format.

To give a feedback about a space environment the user requests to see the available scores to give and the app shows them to it. After selecting the score, the app sends the score to be saved in the database.

Every possible interaction by the Space user to the system can also be performed by the Building user and every interaction these both users can do, can also be done by the Admin, every interaction was only represented once with the intention of not having repetitions.

5. Validation

The testing and validation phase of the development process Is crucial ensure the app is fully functioning, running smoothly and with no bugs. The more the time passes without fixing these bugs bigger it will be the cost to fix them.

According to a Cambridge University study "software bugs cost economy $312 billion per year" and as mentioned previously in this document, the application was being tested while it was being developed.

The tests described in this chapter will ensure the app meets the requirements and runs accordingly.

5.1 Functional Test

In this test we will see if the application fits the requirements described above.

According to the Table 5-1, the application meets the initial requirements mentioned in chapter 3.2.

As described in the development chapter the app has more features than the ones specified previously. The features were asked for the has the app was being developed, and the test results show that these are working as well.

Table 5-1 - Functional Testing

Functional requirement	What happens in the app	Result
Display the building's areas in which there are available environmental data	After log in successfully the user is taken to the layout that shows the available buildings	☺
The user must be able to choose the area he wants to visualize environmental data from	The user is presented with the available floors and spaces in the selected building that have the system	☺
Display updated available environmental data in the selected area	The app presents the last records of every datatype for that space. The layout is updated every 30 seconds	☺
Display feedback's mean value for the selected area	The layout with the data information presents the mean value of the feedback in that area	☺
The user must be able to give feedback for every available area	The user is given the choice to give a feedback for the space. A layout with 5 options is presented.	☺
Display graphs with data values from that day	The user can see the graphs after clicking the corresponding button in the data layout. A different graph is presented for every data type available in that area	☺
Support log in and register of an Admin user	The app askes for log in as soon as the app starts and can connect. The option to register is also available	☺
The Admin is able the configure the tablet to be used by Building users or Space users	The Admin is given the option to change the user. The app then changes the layout according to the type of user chosen.	☺
The Admin can hide some available data from the other users	The Admin has a menu with the option and the layout is presented once requested. Once the user Is changed the data shown Is the correct one.	☺

These tests were performed during the development phase so that any existing bugs could be addressed and fixed has soon as possible.

5.2 Interface Test

The Interface Test assures the application server interface is working well with the web server interface.

This test will check the communication process and the correct display of error messages. Interruptions by the user or the server will also be tested.

Table 5-2 - Interface Testing

Test	What happens in the app	Result
Initiate the app without internet connection	**Error message:** "Fail to connect to IP" User is asked for the IP address to retry communications	☺
Try login and register without internet connection	**Error message:** "Fail to connect. Check the IP" User is asked for the IP address to retry communications	☺
Try to insert invalid data in the "email" field (log in and register)	**Error message:** "Invalid email" The app does not leave the current layout	☺
Try to register with a registered email	**Error message:** "Failed to register"/"Email already registered" The app does not leave the current layout	☺
Try to log in without a registered email	**Error message:** "Failed to register"/"Check email"/"Check password" The app does not leave the current layout	☺
Cut internet connection in any application page	**Error message**: "Failed to connect" The app shows an update button to press when the connections are fixed	☺

As shown In the Table 5-2 the connection between the application and the web server Is running smoothly with the proper error messages when there Is no Internet connection.

These tests were performed in the final stage of the development has they require that the app Is fully running to test all possible variables.

5.3 Test Scenario

The test scenario is a way to illustrate the use case diagram describing the actions the user might take with the application in a simulated scenario.

Imagine subject A, one of the directors of the Centro de Artes in Sines, he arrives at the building and checks his app. Inserts his log in information and enters the app. The option to choose which building he wants to supervise his shown to him, as he only owns rights to the Centro de Artes building information, only this option is shown.

This subject has now the option to see the data information available in every floor and space supervised by the building in the system.

Subject B, an employee of this building has the app in Building mode, with which he can see the environmental data values available in any space of the building supervised by the system. He checks the Temperature in the library space and sees that the temperature is a bit too low, he goes to the graphics by clicking the corresponding button and sees the Temperature values in that space, the graphic shows that there was a decrease of temperature in the last hour.

By contacting the team responsible for the system, the subject B gets to know that there is a problem with the temperature sensor in that space, and that it is being taking care of.

Subject A, knowing about this occurrence, configures the environmental data in that space so that the Temperature values are not shown. After the update, subject B can no longer see the data values for the Temperature in the library.

After the sensor is fixed, subject A configures the environmental data again so that the Temperature values in the library are visible to all the users.

During that day, subject C, a user that goes to the library in the Centro de Artes in Sines to return a book sees a tablet on the wall, the app in this tablet is configured to the Space User mode, so this subject can only see the environmental data values for the space he is in.

The user checks the available data displayed, Temperature, Humidity, Luminosity and Air Quality. He clicks on the graphics button and sees all the records of temperature values of that day, until that time. He can see that there was a drop on the temperature early in the day but he can also see that the temperature raised up to normal levels again in the next hour and that the values were stable from that point on.

The user then goes back to the data display panel and selects the button "Feedback", he is asked how he classifies the environmental conditions in the space he is in and is given five options to choose from, "Very bad", "Bad", "So so", "Good" and "Very good".

Since the user felt comfortable with the environmental conditions in the library, he chose the "Very good" option and returned to the data display layout. In here the user can see that the mean feedback value increased after his feedback was sent.

The scenario described above is not a real experience since, to this date, it was not possible to test the app in a real-life scenario due to the current pandemic.

The purpose to this test is to show in a more practical way the behavior and possible interactions the user can have with the app. Showing how it would impact the users experience inside the building.

5.4 Analysis

According to the tests described above the application is running as expected.

All the requirements described in chapter 3.2 and the added ones explained in chapter 4 are working well and pass all the described tests.

The connection between app and web server were also tested and all the described tests have a satisfactory response from the app.

However, it was not possible yet to test the app in a real scenario of the Ambiosensing field test, specifically in Centro de Arte de Sines. Although it was planned, even according to the Ambiosensing project timeline, due to the COVID-19 quarantine period, most buildings were closed, and the project pilot implementation was postponed.,

6. Conclusion and Future Work

6.1 Conclusion

In this dissertation work the development process of an application was described and explained thoroughly. This app is meant to interact with an Intelligent Supervision system, that is part of the Ambiossensing project financed by the program Portugal 2020.

As explained, this project aims to guarantee the users comfort by maintaining a good environment state inside a building while avoiding energy waste.

The application developed comes as supplementary element for this project and allows the users to verify the environmental data values the system is collecting, see the corrections that are made to these by the Supervision system and give his/her feedback to the system, improving its performance.

By the tests preform and described in the previous chapter we can conclude that the application, as it is now, complies with all given requirements without any errors or bugs detected. A real-life test has yet to be done to exclude all errors/bugs in the system that might still exist.

In conclusion, we can say that the implementation of this app will improve the overall experiences the users can have inside the building.

6.2 Future Work

A future work with this application would be testing the app in a real-life scenario with actual users and real environmental data registered in the database by the Intelligent Supervision system.

Another aspect that would be interesting to study would be how much is the user's feedback affected by the environmental data values the app shows.

Using the test scenario, if a user were to go to the library while the sensor was malfunctioning would the user’s opinion about the environmental condition of the space change if that data value were showing or not.

How much does the hiding reading values feature in the app influences the feedback? I think studying and testing this hypothesis would be interesting in the future.

As this dissertation is about an application, future work will always be related to new updates that surely will be done. Could be an update to improve performance, to correct a late detected bug, or to keep up with new features the Intelligent Supervision System might get in the future.

Technologies' evolves fast and with new methods being study and tested every day this Intelligent supervisor will be improved in the future, as well as the building technology that supports the system will be upgraded and this application that supplements the system will be updated too, to keep up with technology growth.

References

[1] Designing Building, “Designing Buildings Wiki,” *Energy Effic. Build.*, no. Dm, pp. 8–11, 2015.

[2] Designing Buildings, “Human comfort in buildings - Designing Buildings Wiki,” *Wiki, Designing Buildings*, 2019. [Online]. Available: https://www.designingbuildings.co.uk/wiki/Human_comfort_in_buildings. [Accessed: 02-Nov-2020].

[3] K. Parsons, “Thermal comfort in buildings | Multi Comfort - Saint-Gobain,” *Mater. energy Effic. Therm. Comf. Build.*, pp. 127–147, 2010.

[4] UK Essays, “Factors Affecting Human Comfort In Buildings,” 2018. [Online]. Available: https://www.ukessays.com/essays/engineering/human-comfort-in-the-internal-environment.php. [Accessed: 02-Nov-2020].

[5] L. Pérez-Lombard, J. Ortiz, and C. Pout, “A review on buildings energy consumption information,” *Energy Build.*, vol. 40, no. 3, pp. 394–398, 2008, doi: 10.1016/j.enbuild.2007.03.007.

[6] P. H. Shaikh, N. B. M. Nor, P. Nallagownden, I. Elamvazuthi, and T. Ibrahim, “A review on optimized control systems for building energy and comfort management of smart sustainable buildings,” *Renew. Sustain. Energy Rev.*, vol. 34, pp. 409–429, 2014, doi: 10.1016/j.rser.2014.03.027.

[7] M. G. Patterson, “What is energy efficiency? Concepts, indicators and methodological issues,” *Energy Policy*, vol. 24, no. 5, pp. 377–390, 1996, doi: 10.1016/0301-4215(96)00017-1.

[8] Ambiosensing, “Ambiosensing – by Ambiosfera,” 2018. [Online]. Available: https://ambiosensing.com/. [Accessed: 07-Feb-2020].

[9] M. Peluffo, “Defining Today’s Intelligent Building,” *Commscope*, 2015.

[Online]. Available: https://www.commscope.com/Blog/Defining-Todays-Intelligent-Building/. [Accessed: 03-Feb-2020].

[10] Frost & Sullivan, "Convergence of Green and Intelligent Buildings," *Buildings*, 2009.

[11] Green Coast, "A Definitive Guide to Building Energy Management Systems," 2019. [Online]. Available: https://greencoast.org/buGreen Coast. (2019). A Definitive Guide to Building Energy Management Systems. https://greencoast.org/building-energy-management-systems/ilding-energy-management-systems/. [Accessed: 04-Feb-2020].

[12] J. Schmidhuber, "Curious Model-building Control Systems," vol. 2, pp. 1458–1463, 1991.

[13] T. Weng and Y. Agarwal, "From buildings to smart buildings-sensing and actuation to improve energy efficiency," *IEEE Des. Test Comput.*, vol. 29, no. 4, pp. 36–44, 2012, doi: 10.1109/MDT.2012.2211855.

[14] H. Meixner and O. Gassmann, "System Technologies: Sensor Systems in Intelligent Buildings," vol. 7, pp. 483–510, 2001.

[15] A. Monteiro, "Área Departamental de Engenharia Eletrotécnica Energia e Automação Sistemas Inteligentes para melhoria da eficiência energética em edifícios," 2017.

[16] Energy Efficiency & Renewable, "LED Lighting : Department of Energy," 2016. [Online]. Available: https://www.energy.gov/energysaver/save-electricity-and-fuel/lighting-choices-save-you-money/led-lighting. [Accessed: 28-Nov-2020].

[17] D. J. Carnovale, A. Barchowsky, and B. Groden, "Consulting - Specifying Engineer | Saving energy with smart lighting," 2015. [Online]. Available: https://www.csemag.com/articles/saving-energy-with-smart-lighting/. [Accessed: 05-Feb-2020].

[18] Eve Home, "Eve Energy," 2017. [Online]. Available: https://www.evehome.com/en/eve-energy. [Accessed: 28-Nov-2020].

[19] A. Afram and F. Janabi-Sharifi, "Theory and applications of HVAC control systems - A review of model predictive control (MPC)," *Build. Environ.*, vol. 72, pp. 343–355, 2014, doi: 10.1016/j.buildenv.2013.11.016.

[20] J. Contreras, "Facilities Management - Setpoint," 2017. [Online]. Available: https://facilities.ucdavis.edu/blog/word-month-setpoint. [Accessed: 22-Dec-2019].

[21] Advanced Control Corp., "Smart Building Solutions for Energy-Efficient Buildings," 2015. [Online]. Available: http://advancedcontrolcorp.com/blog/2015/03/smart-building-solutions-for-energy-efficient-buildings/. [Accessed: 30-Nov-2019].

[22] V. L. Erickson, M. Á. Carreira-Perpiñán, and A. E. Cerpa, "OBSERVE: Occupancy-based system for efficient reduction of HVAC energy," *Proc. 10th ACM/IEEE Int. Conf. Inf. Process. Sens. Networks, IPSN'11*, pp. 258–269, 2011.

[23] S. Wang and Z. Ma, "Supervisory and optimal control of building HVAC systems: A review," *HVAC R Res.*, vol. 14, no. 1, pp. 3–32, 2008, doi: 10.1080/10789669.2008.10390991.

[24] UC DAVIS, "Facilities Management - TherMOOstat," 2019. [Online]. Available: https://facilities.ucdavis.edu/energy-engineering/thermoostat. [Accessed: 29-Dec-2019].

[25] BuildingIQ, "BuildingIQ Mobile App," *BuildingIQ*, 2020. [Online]. Available: https://buildingiq.com/app/. [Accessed: 04-Feb-2021].

[26] A. I. Dounis and C. Caraiscos, "Advanced control systems engineering for energy and comfort management in a building environment-A review," *Renew. Sustain. Energy Rev.*, vol. 13, no. 6–7, pp. 1246–1261, 2009, doi: 10.1016/j.rser.2008.09.015.

[27] A. Shepherd and W. Batty, "D s d s," vol. 1, no. September, pp. 4–6, 2013.

[28] T. Toshikazu and H. Toshiharu, "A practical application of fuzzy control for an air-conditioning system," *Int. J. Approx. Reason.*, vol. 5, no. 3, pp. 331–348, 1991, doi: 10.1016/0888-613X(91)90016-F.

[29] H. Zhang, A. Davigny, F. Colas, Y. Poste, and B. Robyns, "Fuzzy logic based energy management strategy for commercial buildings integrating photovoltaic and storage systems," *Energy Build.*, vol. 54, pp. 196–206, 2012, doi: 10.1016/j.enbuild.2012.07.022.

[30] R. N. Lea, E. Dohmann, W. Prebilsky, and Y. Jani, "HVAC fuzzy logic zone control system and performance results," *IEEE Int. Conf. Fuzzy Syst.*, vol. 3, no. 1, pp. 2175–2180, 1996, doi: 10.1109/fuzzy.1996.553537.

[31] S. Soyguder, M. Karakose, and H. Alli, "Design and simulation of self-tuning PID-type fuzzy adaptive control for an expert HVAC system," *Expert Syst. Appl.*, vol. 36, no. 3 PART 1, pp. 4566–4573, 2009, doi: 10.1016/j.eswa.2008.05.031.

[32] D. Kolokotsa, D. Tsiavos, G. S. Stavrakakis, K. Kalaitzakis, and E. Antonidakis, "Advanced fuzzy logic controllers design and evaluation for buildings' occupants thermal-visual comfort and indoor air quality satisfaction," *Energy Build.*, vol. 33, no. 6, pp. 531–543, 2001, doi: 10.1016/S0378-7788(00)00098-0.

[33] R. Alcalá, J. Casillas, and A. González, "Tuning Fuzzy Logic Controllers for Energy Efficiency Consumption in Buildings," *EUSFLAT-ESTYLF Jt. Conf*, no. January, pp. 1–4, 1999.

[34] S. A. Kalogirou, "Applications of artificial neural-networks for energy systems," *Appl. Energy*, vol. 67, no. 1–2, pp. 17–35, 2000, doi: 10.1016/S0306-2619(00)00005-2.

[35] H. X. Zhao and F. Magoulès, "A review on the prediction of building energy consumption," *Renew. Sustain. Energy Rev.*, vol. 16, no. 6, pp. 3586–3592, 2012, doi: 10.1016/j.rser.2012.02.049.

[36] S. Kajl, M.-A. Roberge, L. Lamarche, and P. Malinowski, "Evaluation of Building Energy Consumption Based on," *oceedings Clima 2000 Conf. held Brussels, August 30th to Sept. 2nd*, no. 1, pp. 1–10, 1997.

[37] M. Wei, S. H. Hong, and M. Alam, "An IoT-based energy-management platform for industrial facilities," *Appl. Energy*, vol. 164, pp. 607–619, 2016, doi: 10.1016/j.apenergy.2015.11.107.

[38] X. Ma, R. Cui, Y. Sun, C. Peng, and Z. Wu, "Supervisory and energy management system of large public buildings," *2010 IEEE Int. Conf. Mechatronics Autom. ICMA 2010*, pp. 928–933, 2010, doi: 10.1109/ICMA.2010.5589969.

[39] H. Doukas, K. D. Patlitzianas, K. Iatropoulos, and J. Psarras, "Intelligent building energy management system using rule sets," *Build. Environ.*, vol. 42, no. 10, pp. 3562–3569, 2007, doi: 10.1016/j.buildenv.2006.10.024.

[40] L. M. Camarinha-Matos, "SupInt-5-2018." 2018.

[41] R. Vadavi, "Self-healing IT systems: Less of a choice, more a mandatory requirement | Blog| Mindtree," *Mindtree*, 2019. [Online]. Available: https://www.mindtree.com/blog/self-healing-it-systems-less-choice-more-mandatory-requirement. [Accessed: 04-Mar-2021].

[42] Alessandro, "Metodologia de Desenvolvimento de Software," *DevMedia*, pp. 1–30, 2006.

[43] E. Lúcia De Moura and M. Moreira, "METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE," 2015.

[44] smartsheet, "A Complete Guide to the Waterfall Project Method | Smartsheet," *smartsheet*, 2020. [Online]. Available: https://www.smartsheet.com/content-center/best-practices/project-management/project-management-guide/waterfall-methodology. [Accessed: 23-Sep-2020].

[45] S. Ken and S. Jeff, "Scrum - Quick Guide - Tutorialspoint," *Tutorials Point*, 2013. [Online]. Available: https://www.tutorialspoint.com/scrum/scrum_quick_guide.htm. [Accessed: 18-Nov-2020].

[46] UNIVESP, "Engenharia de Software - Aula 01 - Modelos de processo de software e atividades de software - YouTube," 2017. [Online]. Available:

https://www.youtube.com/watch?v=kO1PSkzTsYc&ab_channel=UNIVESP. [Accessed: 23-Sep-2020].

[47] P. Kruchten, "Dimensions of the wind turbine tower. | Download Scientific Diagram," *Research Gate*, 2000. [Online]. Available: https://www.researchgate.net/figure/Two-Dimensions-of-the-RUP_fig2_220018149. [Accessed: 18-Nov-2020].

[48] UNIVESP, "Engenharia de Software - Aula 02 - Processo unificado - YouTube," 2017. [Online]. Available: https://www.youtube.com/watch?v=o-09AFlJZTo&t=773s&ab_channel=UNIVESP. [Accessed: 23-Sep-2020].

[49] Institute of Electrical and Electronics Engineers, "IEEE Standard Glossary of Software Engineering Terminology," *Office*, vol. 121990, no. 1, p. 1, 1990.

[50] Codificar, "Requisitos funcionais e requisitos não funcionais, o que são?," 2017. [Online]. Available: https://codificar.com.br/requisitos-funcionais-nao-funcionais/. [Accessed: 10-Sep-2020].

[51] QRA Team, "Functional vs non-functional requirements: The definitive guide," *Qra*, 2019.

[52] Nishadha, "UML Diagram Types | Learn About All 14 Types of UML Diagrams," *creately*, 2020. [Online]. Available: https://creately.com/blog/diagrams/uml-diagram-types-examples/. [Accessed: 08-Oct-2020].

[53] Kirill Fakhroutdinov, "Unified Modeling Language (UML) description, UML diagram examples, tutorials and reference for all types of UML diagrams - use case diagrams, class, package, component, composite structure diagrams, deployments, activities, interactions, profiles, etc.," *The Unified Modeling Language*, 2018. [Online]. Available: https://www.uml-diagrams.org/. [Accessed: 08-Oct-2020].

[54] Neol Ceta, "All You Need to Know About UML Diagrams: Types and 5+ Examples," *Tallyfy*, 2015. [Online]. Available: https://tallyfy.com/uml-diagram/. [Accessed: 08-Oct-2020].

[55] Veracode, "What is IDE or Integrated Development Environments? | Veracode," *Veracode*, 2020. [Online]. Available: https://www.veracode.com/security/integrated-development-environment. [Accessed: 17-Oct-2020].

[56] R. Hat, "O que é IDE? - Ambiente de Desenvolvimento Integrado," 2020. [Online]. Available: https://www.redhat.com/pt-br/topics/middleware/what-is-ide. [Accessed: 17-Oct-2020].

[57] M. Ashraf, "Best IDEs for Mobile App Development | Our Top 5 Picks," *App Verticals*, 2020. [Online]. Available: https://www.appverticals.com/blog/best-integrated-development-

environments/. [Accessed: 17-Oct-2020].

[58] M. Kataria, “Ending the Native vs Cross Platform Debate,” *SimForm*, 2016. [Online]. Available: https://www.simform.com/native-vs-cross-platform-development/. [Accessed: 17-Oct-2020].

[59] Austin, “Cross-Platform Development: Native vs. Native Cross-Platform vs. Hybrid Cross-Platform - Austin Howard Tech,” *Austin Howard Tech*, 2019. [Online]. Available: https://austinhoward.tech/cross-platform-development-comparison/. [Accessed: 17-Oct-2020].

[60] A. Lazar, “The battle of the IDEs - JAXenter,” *jaxenter*, 2018. [Online]. Available: https://jaxenter.com/battle-ides-150122.html. [Accessed: 17-Oct-2020].

[61] N. Maldonado, “What is the pros and cons of Netbeans IDE for web development? - Quora,” *Quora*, 2016. [Online]. Available: https://www.quora.com/What-is-the-pros-and-cons-of-Netbeans-IDE-for-web-development. [Accessed: 17-Oct-2020].

[62] P. Ganesan, “What is XAMPP? and How to Install XAMPP on Local Computer?,” *Wpblogx*, 2017. [Online]. Available: https://www.wpblogx.com/what-is-xampp/. [Accessed: 21-Oct-2020].

[63] S. Balkhi, “What is Apache? - What is a Web Server?,” *Biginners Guide for Wordpress*, 2009. [Online]. Available: https://www.wpbeginner.com/glossary/apache/. [Accessed: 21-Oct-2020].

[64] JReport, “3-Tier Architecture: A Complete Overview,” 2018. [Online]. Available: https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/. [Accessed: 28-Apr-2020].

[65] Izenda Editorial Staff, “5 Benefits of a 3-Tier Architecture - Izenda,” *Analytics, Embedded BI*, 2017. [Online]. Available: https://www.izenda.com/5-benefits-3-tier-architecture/. [Accessed: 28-Apr-2020].

[66] Android, “Introduction to Activities | Android Developers,” *Android Developers*, 2015. [Online]. Available: https://developer.android.com/guide/components/activities/intro-activities. [Accessed: 11-Nov-2020].

[67] André Servidoni, “Android Basics - Activities & Fragments,” *Online Article*, 2017. [Online]. Available: https://blog.avenuecode.com/android-basics-activities-fragments. [Accessed: 11-Nov-2020].

[68] Community, “android - Dilemma: when to use Fragments vs Activities: - Stack Overflow,” *stackoverflow*, 2017. [Online]. Available: https://stackoverflow.com/questions/20306091/dilemma-when-to-use-fragments-vs-activities. [Accessed: 14-Nov-2020].

[69] N. Shah, “How to Fetch Data from Database in PHP and display in HTML Table?,” *CodeAndCourse.com*, 2020. [Online]. Available: https://www.geeksforgeeks.org/how-to-fetch-data-from-localserver-database-and-display-on-html-table-using-php/?ref=lbp. [Accessed: 15-Nov-2020].

[70] w3schools, “SQL INNER JOIN Keyword,” 2016. [Online]. Available: https://www.w3schools.com/sql/sql_join_inner.asp. [Accessed: 15-Nov-2020].

[71] W3schools Group, “SQL GROUP BY Statement,” 2019. [Online]. Available: https://www.w3schools.com/sql/sql_groupby.asp. [Accessed: 15-Nov-2020].

[72] MaRS Startup Toolkit, “Product development: Waterfall Model | Software development methodology | Entrepreneur’s Toolkit,” *learn.marsdd.com*, 2018. [Online]. Available: https://learn.marsdd.com/article/product-development-the-waterfall-methodology-model-in-software-development/. [Accessed: 18-Nov-2020].