

POLYTECHNIC INSTITUTE OF CÁVADO AND AVE

Smart Campus - Residence

Bruna Raquel Novera Macieira

Ricardo Daniel da Silva Teixeira



Bachelor in Computer Systems Engineering

Supervisor: Prof. Eduardo Peixoto

January 10, 2023

Smart Campus - Residence

Bruna Raquel Novera Macieira

Ricardo Daniel da Silva Teixeira

Bachelor in Computer Systems Engineering

January 10, 2023

Abstract

The concern with process optimisation and cost reductions are some of the main focuses that occur when managing a university.

The aim of this project is to build a smartphone application (Android) that can control the environment someone (in this case, a student) is in, such as turning on/off the lights, controlling the temperature and other aspects, all at just one click of a button.

This project not just only facilitates the life of a regular student, but is also built thinking of students with disabilities, since the handling/control of the environment is easier.

*“You should be glad that bridge fell down.
I was planning to build thirteen more to that same design”*

Isambard Kingdom Brunel

Contents

List of Figures	v
List of Tables	vi
1 Introduction	1
1.1 Contextualization	1
1.1.1 Who's the customer	1
1.1.2 Who are the users	1
1.1.3 Improvements of the project on daily life of users	1
1.1.4 Stakeholders and interested parts	1
1.2 Deliveries	2
2 Deliver 1	3
2.1 Contextualization	3
2.1.1 Business Canvas Model	3
2.1.2 Budget	5
2.1.3 Functional Requirements	5
2.1.4 Non-Functional Requirements	6
2.2 Wireframes	6
3 Deliver 2	9
3.1 Product Backlog	9
3.1.1 User	9
3.1.2 Cleaning staff	10
3.1.3 Admin	10
3.1.4 Guest	10
3.2 Use Cases Diagram	10
3.3 Entity-Relation Diagram	13
4 Delivery 3	14
4.1 State Diagram	14
4.2 Activities Diagram	14
4.3 Sequence Diagram	14
4.4 Mockups	14
5 Information System Integration	22
5.1 Architecture of the System	22
5.1.1 Database	22
5.1.2 Technology of Infrastructure	24

5.1.3	Security- Roles	24
5.1.4	Integration levels	25
5.1.5	Problem Solution	25
5.1.6	Endpoints	25
5.1.7	Code Implementation	25
5.1.8	Conclusion	26
5.1.9	Bibliography	27
6	Embedded Systems	28
6.1	Contextualization and Motivation	28
6.2	Architecture of the System	28
6.2.1	Justification of the measures taken	29
6.2.2	Conclusion	30
6.2.3	Bibliography	30
	References	31
A	Group organisation dossier	32
A.1	Internal Rules	32
A.1.1	Scope	32
A.1.2	Workgroup constitution	32
A.1.3	Meetings	32
A.1.4	Internal evaluation	33
A.2	Timetable	33
A.3	Meeting records	33
A.3.1	Meeting 1	33
A.3.2	Meeting 2	34
A.3.3	Meeting 3	34
A.3.4	Meeting 4	34
A.3.5	Meeting 5	35
A.3.6	Meeting 6	35
A.3.7	Meeting 7	35
A.3.8	Meeting 8	36
A.3.9	Meeting 9	36
A.3.10	Meeting 10	37
A.3.11	Meeting 11	37
A.3.12	Meeting 12	37
A.3.13	Meeting 13	38
A.4	Internal Evaluation	38
A.4.1	Deliveries 1 and 2	38
A.4.2	Delivery 3	38
A.4.3	Delivery 4	39

List of Figures

2.1	Business Model Canvas as described in page 3	5
2.2	Wireframes from Login pages and Room Control UI	7
2.3	Wireframe from Other Funcionalities UI	8
3.1	Use Case Diagram 1 from the system	11
3.2	Use Case Diagram 2 from the system	12
3.3	Entity-Relation Diagram from the system	13
4.1	State Diagram from the system	15
4.2	Activities Diagram from the system	15
4.3	Sequence Diagram from the system	16
4.4	Initial Page mockup	16
4.5	Guest Login mockup	17
4.6	Student Login mockup	18
4.7	Guest Access mockup	19
4.8	Student's Room Control mockup	20
4.9	Cleaning and Guest Access Management mockup	21
5.1	Document-based MongoDB of the entity Cleaning Appointment	22
5.2	Document-based MongoDB of the entity Guest	22
5.3	Document-based MongoDB of the entity Person	23
5.4	Document-based MongoDB of the entity Room	23
5.5	Document-based MongoDB of the entity Student	23
5.6	Architecture of the System	24
6.1	Circuit of the residence system	29
6.2	Circuit 3D of the residence system	29
A.1	Timetable 1 - Meetings and Work Planning	33
A.2	Timetable 2 - Meetings and Work Planning	33

List of Tables

Abbreviations and symbols

AI	Artificial Intelligence
API	Application Programming Interface
IPCA	Instituto Politécnico do Cávado e do Ave - (Polytechnic Institute of Cávado and Ave)
LCD	Liquid-Crystal Display
QR	Quick Response (code)

Chapter 1

Introduction

This project was conceived as a joint of the curricular units “Projeto Aplicado”, “Sistemas Embebidos e de Tempo Real”, “Integração de Sistemas de Informação”, “Programação de Dispositivos Móveis” and “Inteligência Artificial”.

1.1 Contextualization

The aim of this project is to be applied on IPCA’s new university residencies, as a way of bringing technology to improve communication between all the involved ones.

1.1.1 Who’s the customer

The customer is IPCA, represented by teacher Eduardo Peixoto, our mentor of “Projeto Aplicado”.

1.1.2 Who are the users

The users are all the involved parts when it comes to a student’s room, such as the cleaning staff, the admin, the student’s guests (in case they want to visit them) and the student themselves.

1.1.3 Improvements of the project on daily life of users

This project brings lots of benefits, such as: enabling the student to control the environment they’re in; improving the communication between the student and the cleaning staff, throughout the LCD display built on every door, telling the staff if they desire their room to be cleaned or not; providing an easier way for disabled people to control the environment they’re in, through AI and distance-of-a-click buttons; bringing IPCA to the next level when it comes to technology.

1.1.4 Stakeholders and interested parts

The main stakeholder of this project is IPCA and its partners.

1.2 Deliveries

- All documentation including: analyze the problem, contextualization, requirements, product backlog, diagrams, mockups and workgroup documentation
- Front-end, mobile application
- Back-end, web API
- Methodology **Scrum**(?)¹ will be used for project planning
- The chosen working repository is GitHub, where the project's version control will also be done
- This document is organised and described in the document structure
- Group organisation dossier will be on Appendix of this document

Problem identification:

- Deliver 1, analysis of requirements and modelling. Specification of the functional and non-functional requirements of the system, complete backlog and initial planning of sprint 1
- Deliver 2, ER and UC diagrams, plus additional Product Backlog and important information
- Deliver 3, Sequence and Activities diagrams, plus mockups and additional information
- Deliver 4, poster plus first code
- Deliver 5, final code
- (ISI) Information Systems Integration, all the information in this project related to this curricular unit

¹ Scrum is a framework that helps teams work together.

Chapter 2

Deliver 1

In this initial phase of the project, the deliverable pieces are as follows:

- Business Canvas Model
- Functional Requirements
- Non-Functional Requirements
- Wireframe
- Budget
- Group organisation dossier
 - Internal rules of procedure
 - Timetable
 - Internal evaluation system

2.1 Contextualization

2.1.1 Business Canvas Model

- Key partners:
 - IPCA, represented by teacher Eduardo Peixoto
 - From our partners, we expect to receive financial and moral support
 - Key activities performed by them will give us clear information about necessities and desires, so the project can be quickly developed
 - Main motivation to work with them is their ability to help us grow and freedom of choice; they believe us enough to hand us the project
- Key activities:

- Extensive safety testing to ensure the best performance of the application
- Automation using AI
- Key resources:
 - Human capital in engineers (team and installation technicians)
 - Capital resources in manufacturing equipment, frameworks and licenses
 - Raw materials such as sensors and power units
- Value propositions:
 - To provide an easiest way to the management of Campus Residence
 - Use of technology to increase the energy efficiency
- Customer relationship:
 - Terrific customer and maintenance services
 - Customer service contacts and direct maintenance appointments
- Channels:
 - Application integration on all university residences.
- Customer segments:
 - Corporate looking for an easy and affordable way to save money and resources on electricity using AI
 - Residences increased safety
 - University students and employees in need of a better way to keep everything under control
- Cost structure:
 - Primary costs are related to material supplies, licenses and frameworks
 - Secondary costs are related to installation of above mentioned products on residence rooms and spaces
 - Sourcing through key partners allows us to save on raw products costs and our specific-driven application allows us to use already existing products
- Key resources:
 - Primary revenue stream comes from sale of the product
 - Secondary revenue stream includes installation on all university residences and their partners, not just the main one
 - Tertiary revenue stream includes maintenance and customer service

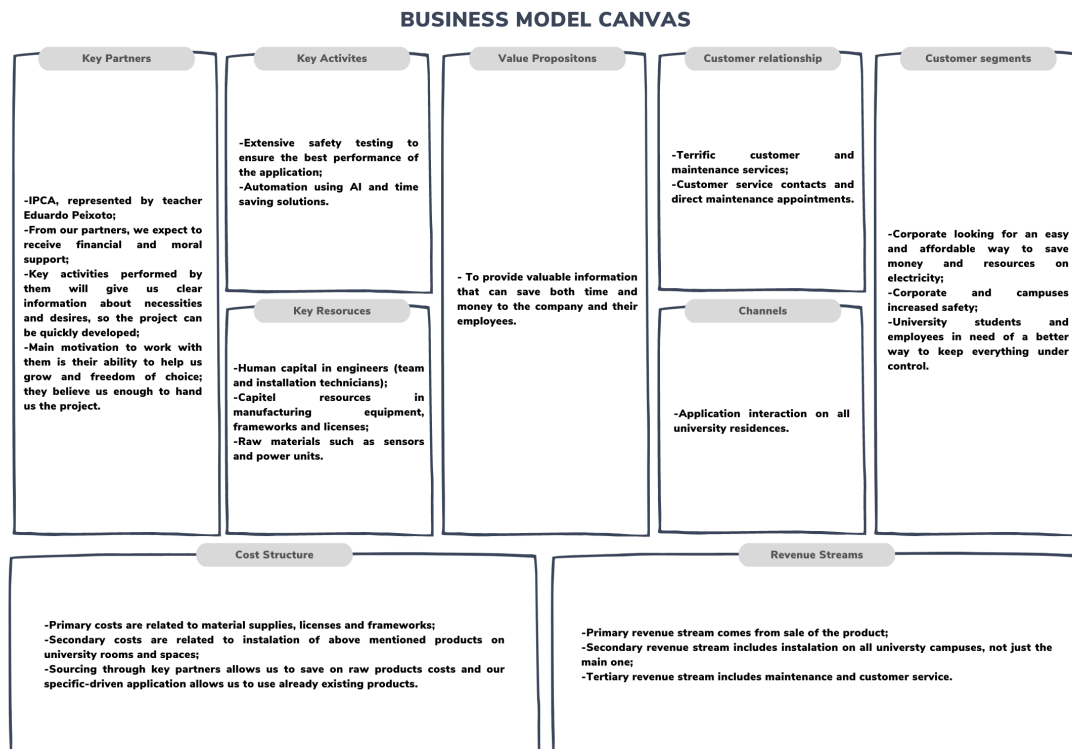


Figure 2.1: Business Model Canvas as described in page 3

2.1.2 Budget

At this moment it is not yet possible to determine the budget of the project because:

- Technical elements of the project are still missing and will be identified later
- It will also be difficult to determine the time needed for the development of the project because the elements of the project do not all have the same response capacity in fulfilling the tasks

2.1.3 Functional Requirements

- The mobile app must have four types of users: the administrator, the users (students), the cleaning staff and the guests.
- Only the authorized users must be able to use the mobile app.
- All users must be able to login on the system using the mobile app.
- Administrator must be able to register or manage the users on the mobile app.
- Student must be able to change the temperature using the mobile app.
- Student must be able to turn on/off the lights using the mobile app.

- Cleaning staff must be able to enter the users' rooms using their RFID card.
- Cleaning staff's chores/rooms should all be organized by sections/floors.
- Cleaning staff must change the room status as soon as they're done with the room.
- Student must be notified by push notifications when the cleaning staff enters and leaves their room.
- Guest must be able to enter their room unlocking the door using the mobile app.
- Student must be able to send a request to their guests so they can create an account and, after registration, have the permission in their account.
- Student must be notified by push notifications when their guests unlock or lock the room.
- Student must have access to their guests permissions.
- Student must have access to the status of his room and change it.

2.1.4 Non-Functional Requirements

- The API must support receiving, sending and processing information from all the sensors and all clients using the mobile app
- The mobile app must be user-friendly
- The mobile app and all communications must be secure and protected
- The mobile app must run on Android smartphones

2.2 Wireframes

The wireframes represent a basic layout of the project.

These wireframes represent the home page where we can choose the option to login. A login page for each option. As well as a page for controlling the room and another for scheduling cleaning and giving access to guests.

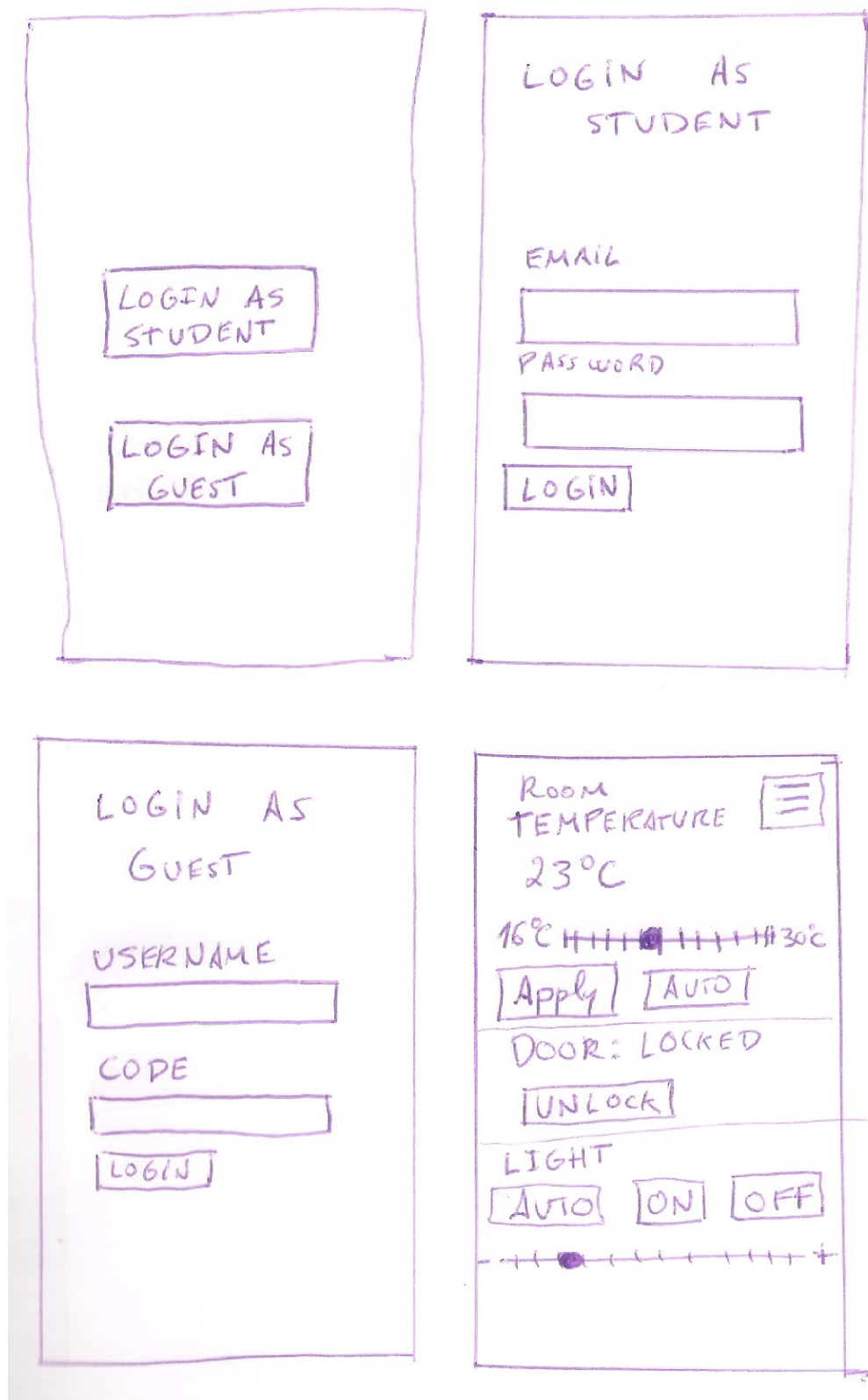


Figure 2.2: Wireframes from Login pages and Room Control UI

A hand-drawn wireframe of a mobile application interface, enclosed in a rectangular border. The interface is organized into several sections:

- CLEAN MY ROOM**: A title at the top, followed by a date input field with a placeholder format of `--/--/-- --:--`.
- APPOINTMENTS**: A section header below the date field.
- DATE**: A label for a date input field, which includes a small calendar icon on the right.
- GENERATE CREDENTIALS FOR GUEST ACCESS**: A multi-line text label.
- USERNAME**: A label for a text input field.
- CODE:**: A label for a text input field.
- GENERATE** and **EDIT**: Two rectangular buttons positioned side-by-side at the bottom of the form.

Figure 2.3: Wireframe from Other Funcionalities UI

Chapter 3

Deliver 2

In this chapter, it's gonna be presented the Product Backlog, along with the Entity-Relation and Use Cases Diagram.

3.1 Product Backlog

Actors: Cleaning Staff, User, Admin, Guest

3.1.1 User

- as a Student, I want to login so I can control my room through mobile app
- as a Student, I want to logout so I don't have the session open on the mobile app
- as a Student, I want to change my profile so I can update my personal info
- as a Student, I want to turn on/off the lights so I can control the lights on the mobile app
- as a Student, I want to change the room status to "do not disturb" or "away", so people act accordingly
- as a Student, I want to change the room temperature, so I can feel more comfortable
- as a Student, I want to unlock the door through my RFID card or mobile app so I can access my room
- as a Student, I want to give permission to someone to enter my room, so a friend or family can access my room
- as a Student, I want to inactivate the AI feature so I can control the room acclimatization myself
- as a Student, I want to activate the AI feature so the room will be self-manageable/self-acclimatized

3.1.2 Cleaning staff

- as a Cleaning Person, I want to open the door using the RFID card so I can clean the room
- as a Cleaning Person, I want to change the room status (by scanning the card) so they can know that the room was cleaned
- as a Cleaning Person, I want to login, so I can have access to the rooms that I have to clean and register all work done
- as a Cleaning Person, I want to logout, so I don't have the session opened on the mobile app

3.1.3 Admin

- as the Admin, I want to register a User, so the User can have access and control to their room
- as the Admin, I want to check a User, so I can use the User information when needed
- as the Admin, I want to edit/manage a User, so the system will have updated data
- as the Admin, I want to receive a push notification when there's something unusual happening in a room so I can take action

3.1.4 Guest

- as a Guest, I want to login, so I can access the room from my friend or family with permission, even if they're not there
- as a Guest, I want to logout, so I don't have the session opened on the mobile app
- as a Guest, I want to open the door through the mobile app, so I can access the room with permission

3.2 Use Cases Diagram

The presented Use Case Diagram describes the main functionalities of each actor (including the system). For a better understanding of the diagram, it's given the Use Case Slices right above it, plus the Product Backlog.

- Login- Both user and guest can login into the system;
- Give permission- The user can give permission to a guest, so they can enter their room;
- Monitor space- The user must be able to monitor their space when they're not around (such as temperature, humidity, check if there's strange activity going on, etc);

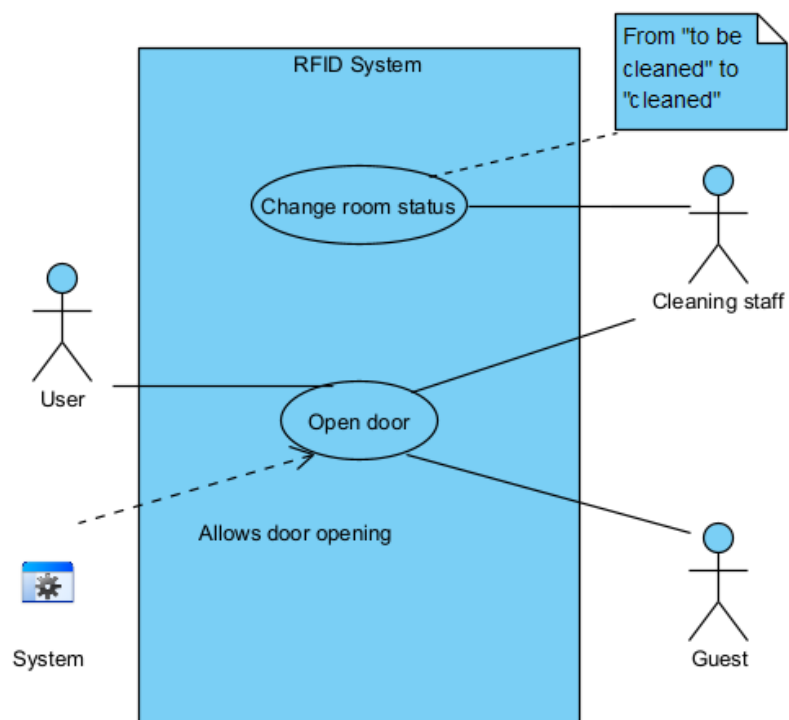


Figure 3.1: Use Case Diagram 1 from the system

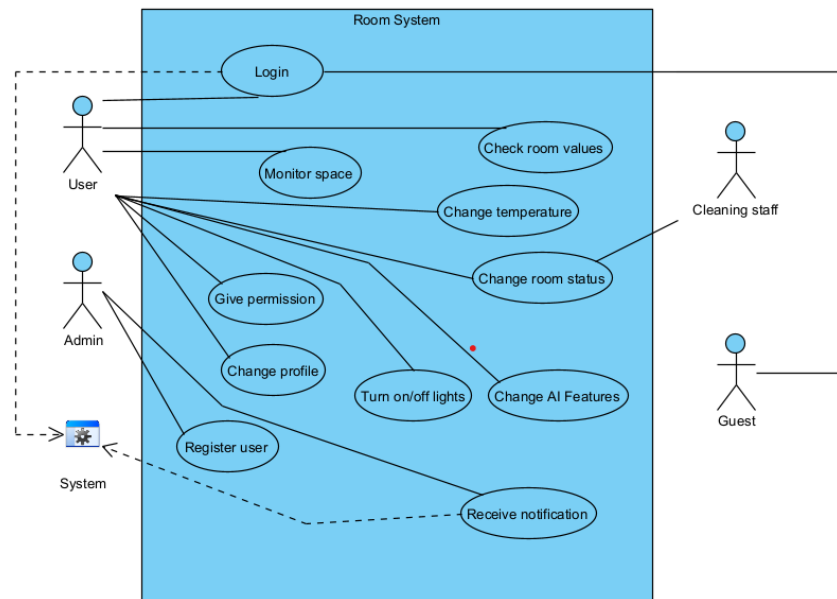


Figure 3.2: Use Case Diagram 2 from the system

- Change profile- The admin must be able to manage/edit the user's profile if they need to, but the user, while active, must also be able to manage their own profile;
- Open door- The user must be able to open their own door; the guest and the cleaning staff must be able to open a door if they have permission for it;
- Turn on/off lights- The user must be able to turn on/off the lights of their own room, both manually and through the app from their phone;
- Change temperature- The user must be able to change the temperature of their own room through the app from their phone;
- Check room values- The user must be able to check the values of their room through the app;
- Change room status- The user must be able to change their room status to "needs to be cleaned" or "do not disturb" when they want to;
- Receive notification- Both admin and the user must receive a notification when there's strange activity detected by the sensors;
- Change AI features- The user must be able to unable/enable the AI features if they want to (like change the light/temperature automatically);
- Register user- The admin must be able to register a user

3.3 Entity-Relation Diagram

As seen on the figure 3.3:

The data model from the system is composed of the following entities:

- User, as known as Student
- Cleaning Person
- Guest
- Sensor
- Room
- Cleaning Appointment

In this system we have the rooms that will be accessed by the users authorized (students, guests or cleaning person) for each room in certain time. Each student will be able to schedule cleanings. And the cleaning will be carried out by a cleaning person. All rooms have sensors to retrieve information about the rooms. Each room can accommodate more than one student.

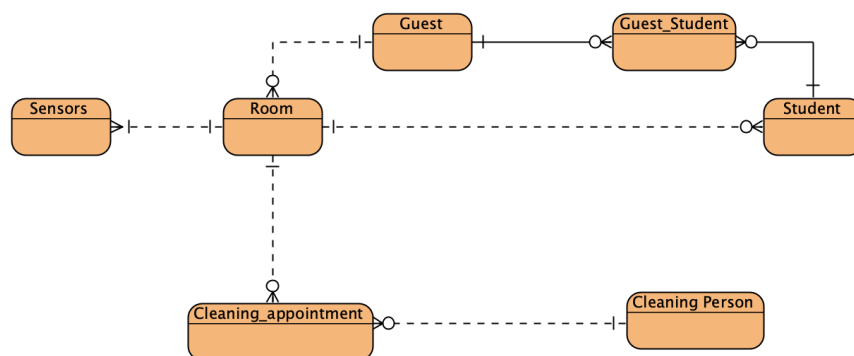


Figure 3.3: Entity-Relation Diagram from the system

Chapter 4

Delivery 3

In this chapter, it's gonna be presented the State Diagram, along with the Activities Diagram, Sequence Diagram and Mockups.

4.1 State Diagram

The State Diagram on the figure 4.1 describes the states of the RFID door system, illumination and cleaning of the room. Whenever someone accesses the interior of the room, its status must be Unlocked, only then can the lights be turned on. Whenever a person leaves its interior, be it a student, a guest, or a cleaning staff, the status must return to Locked. This way you can efficiently help to control the lights, in case we forget to switch them off.

4.2 Activities Diagram

The presented Activities Diagram on the figure 4.2 describes the dynamic aspects of the system. It is an essential advanced flow chart that models the flow from one activity to another activity.

4.3 Sequence Diagram

The Sequence Diagram on the figure 4.3 represents the interactions between objects in the sequential order that those interactions occur.

4.4 Mockups

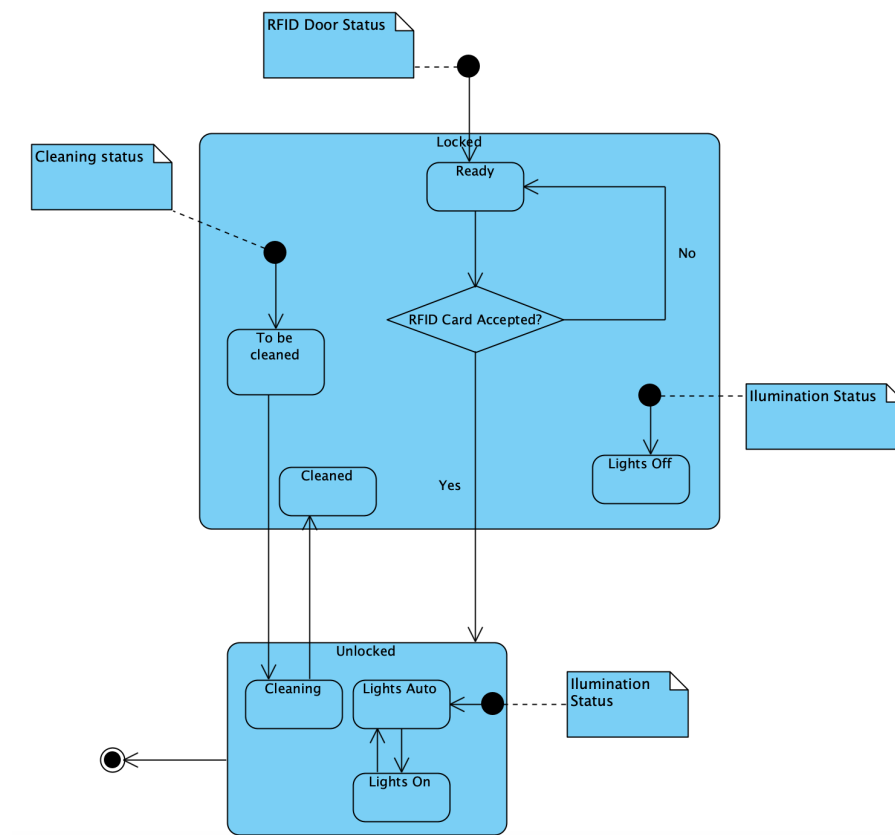


Figure 4.1: State Diagram from the system

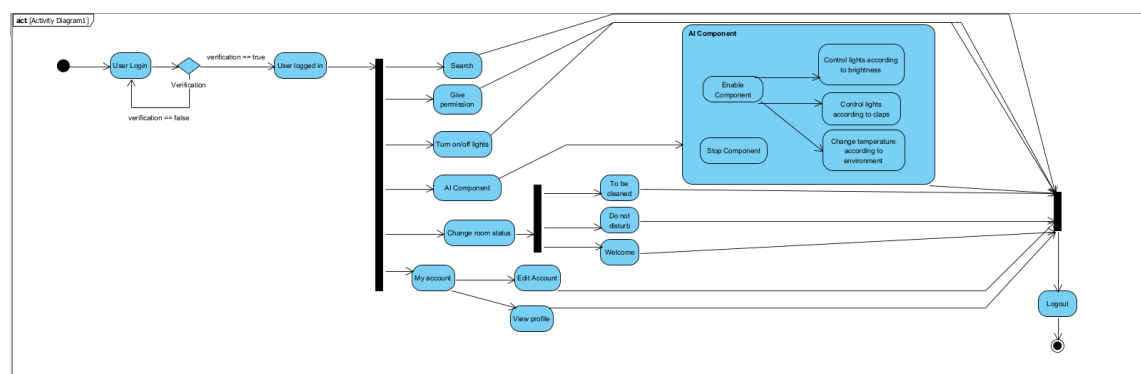


Figure 4.2: Activities Diagram from the system

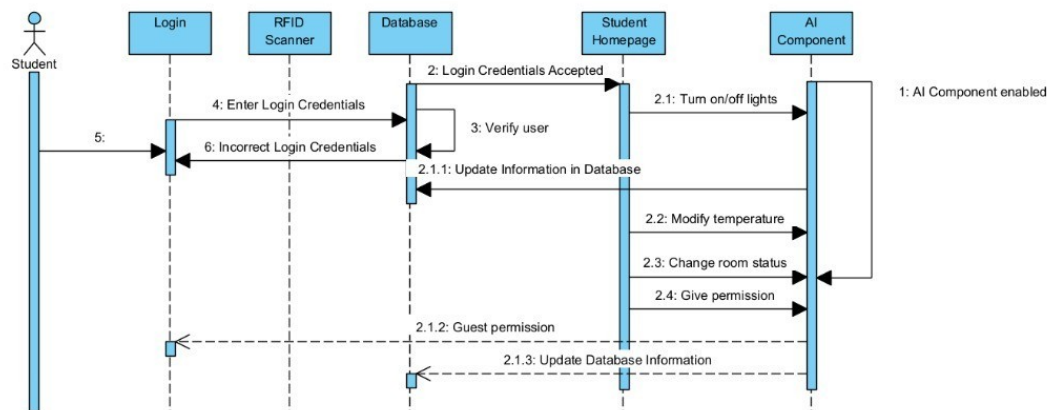


Figure 4.3: Sequence Diagram from the system

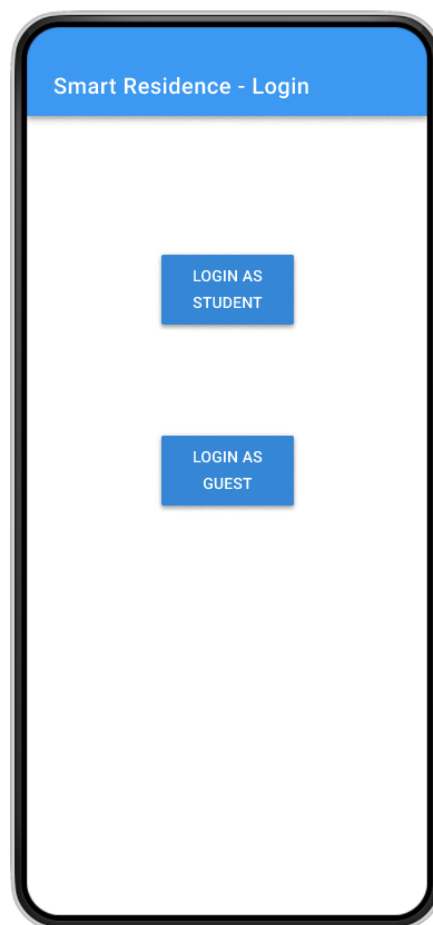
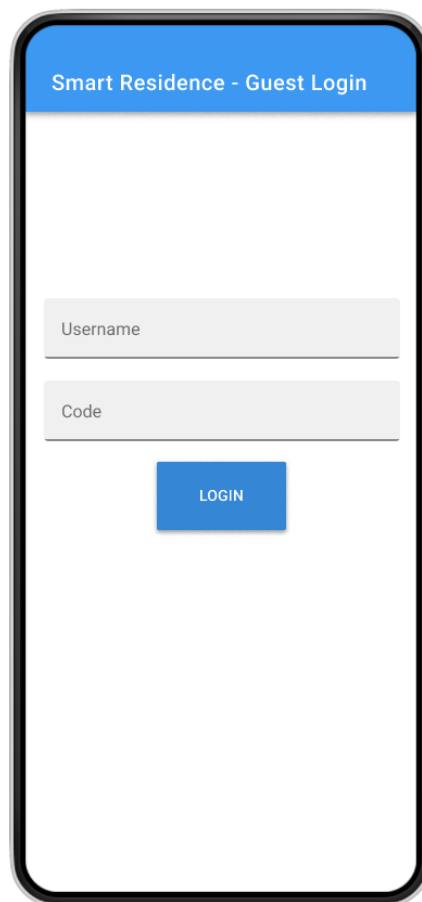
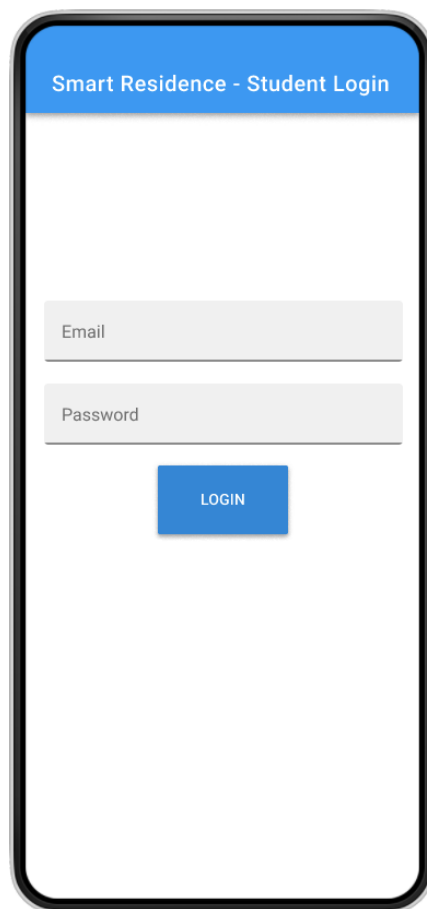


Figure 4.4: Initial Page mockup



A mobile app mockup for 'Smart Residence - Guest Login'. The screen has a blue header bar with the text 'Smart Residence - Guest Login'. Below the header, there are two light gray input fields. The first field is labeled 'Username' and the second field is labeled 'Code'. Below these fields is a blue button with the text 'LOGIN' in white capital letters. The entire mockup is framed by a black border representing a smartphone.

Figure 4.5: Guest Login mockup



A mobile app mockup for 'Smart Residence - Student Login'. The interface features a blue header bar with the title 'Smart Residence - Student Login'. Below the header, there are two light gray input fields: the first is labeled 'Email' and the second is labeled 'Password'. Centered below these fields is a blue rectangular button with the text 'LOGIN' in white capital letters. The entire mockup is framed by a black border representing a smartphone screen.

Figure 4.6: Student Login mockup

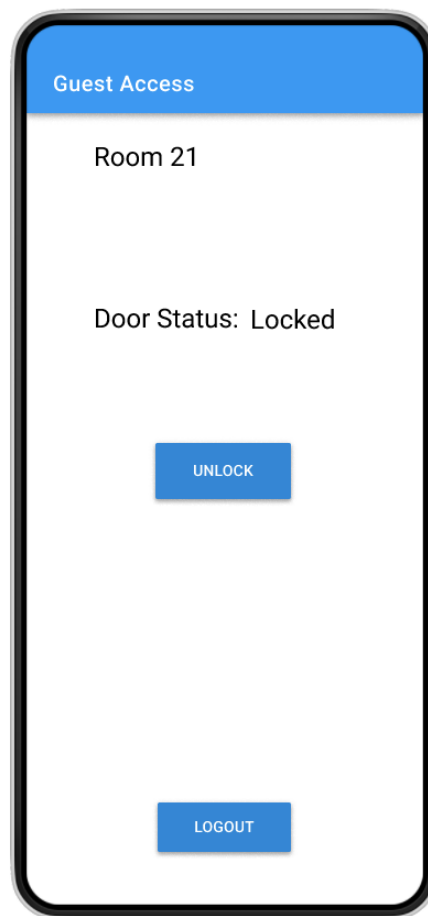


Figure 4.7: Guest Access mockup

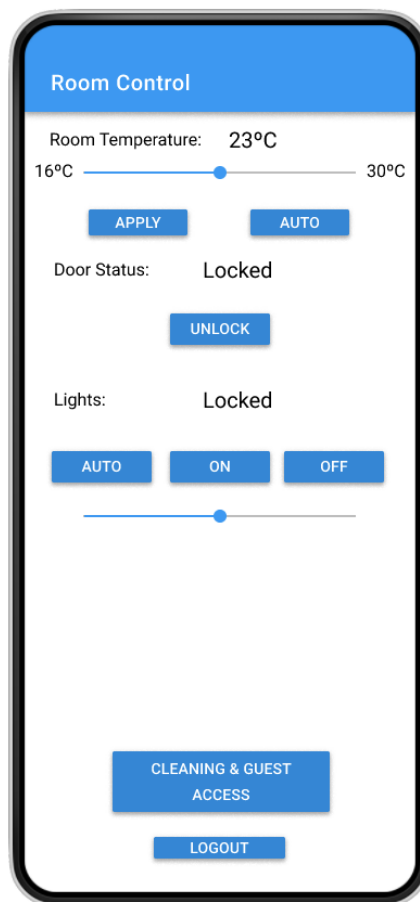
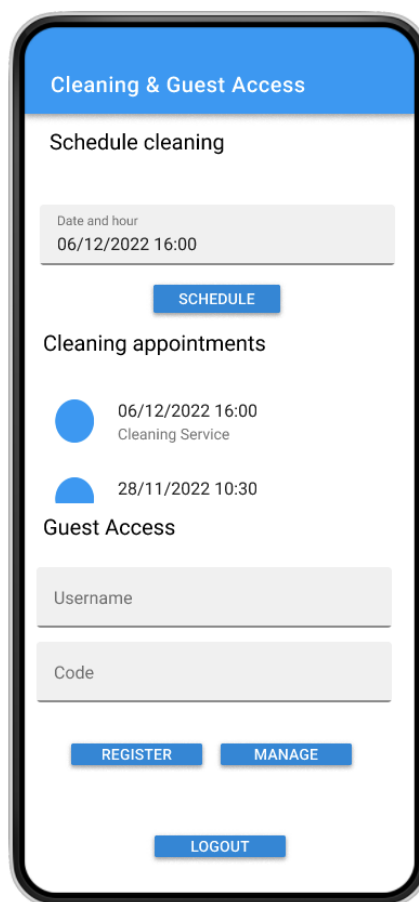


Figure 4.8: Student's Room Control mockup



The mockup displays a mobile application interface for 'Cleaning & Guest Access'. It features a blue header bar with the title 'Cleaning & Guest Access'. Below the header, the 'Schedule cleaning' section includes a date and time input field set to '06/12/2022 16:00' and a blue 'SCHEDULE' button. The 'Cleaning appointments' section lists two entries: '06/12/2022 16:00 Cleaning Service' with a full blue circle icon, and '28/11/2022 10:30' with a half-blue circle icon. The 'Guest Access' section contains input fields for 'Username' and 'Code', followed by 'REGISTER' and 'MANAGE' buttons, and a 'LOGOUT' button at the bottom.

Cleaning & Guest Access

Schedule cleaning

Date and hour
06/12/2022 16:00

SCHEDULE

Cleaning appointments

06/12/2022 16:00
Cleaning Service

28/11/2022 10:30

Guest Access

Username

Code

REGISTER MANAGE

LOGOUT

Figure 4.9: Cleaning and Guest Access Management mockup

Chapter 5

Information System Integration

5.1 Architecture of the System

As mentioned in previous chapters, this project revolves around the idea of implementing an application for our university campus' residence, making it easier for students and other people to control it.

This chapter is based on the API, developed in C# language, that will allow the software components to communicate with each other.

5.1.1 Database

Despite the ER Diagram of the system on the figure 3.3, the database of the system is no-SQL and is supported by MongoDB document-based with the data structures examples of the figures 5.1, 5.2, 5.3, 5.4 e 5.5.

SmartResidence.Cleaning Appointment

```
_id: ObjectId('63a4e56787b360fc21e6b059')
room_id: ObjectId('63a4e57787b360fc21e6b05a')
cleaning_person: ObjectId('63a4e58987b360fc21e6b05b')
status: 0
date: 2023-01-12T15:00:00.000+00:00
```

Figure 5.1: Document-based MongoDB of the entity Cleaning Appointment

SmartResidence.Guest

```
_id: ObjectId('63a4e4df87b360fc21e6b058')
username: "friend021"
code: "43jfoiefjejf09e"
status: 1
```

Figure 5.2: Document-based MongoDB of the entity Guest

SmartResidence.Cleaning Person

```
_id: ObjectId('63a4e3ce63ae82448848478d')
name: "Gorete Almeida"
rfid_card: "28j239813"
```

Figure 5.3: Document-based MongoDB of the entity Person

SmartResidence.Room

```
_id: ObjectId('63a4df9bec9561ead65c39c3')
> list_of_guests_allowed: Array
> list_of_room_students: Array
  light_status: 1
  light_intensity: 65
  door_status: 1
> next_cleaning_appointments: Array
> past_cleaning_appointments: Array
  room_status: 3
  ac_status: 0
  climatization_temperature_def: 22.5
  last_checked_temperature: 23.2
```

Figure 5.4: Document-based MongoDB of the entity Room

SmartResidence.Student

```
_id: ObjectId('63a4e1f2ec9561ead65c39c8')
name: "Pedro Miguel"
birth_date: 2000-11-05T00:00:00.000+00:00
student_nr: "32049"
email: "a32049@alunos.ipca.pt"
address: "Rua de Barcelinhos, 294"
postcode: "4750-144"
city: "Barcelos"
rfid_card: "o32jf0938fj349f8"
```

Figure 5.5: Document-based MongoDB of the entity Student

5.1.2 Technology of Infrastructure

- The base technology used is .NET 7 as Web API
- All the external communications of the API use REST architecture except Database operations
- Most of the API consumption use Authorization and Authentication with JSON Web Tokens
- The API is consumed by an application connected to an embedded system and by a mobile application (Android)

The REST API used in this project works through the solicitation of data from the client to the server and the server will use those entries to return the output data. In other words, the client can run CRUD functions (such as GET, PUT, DELETE, etc) to access the server's data. They communicate through HTTP.

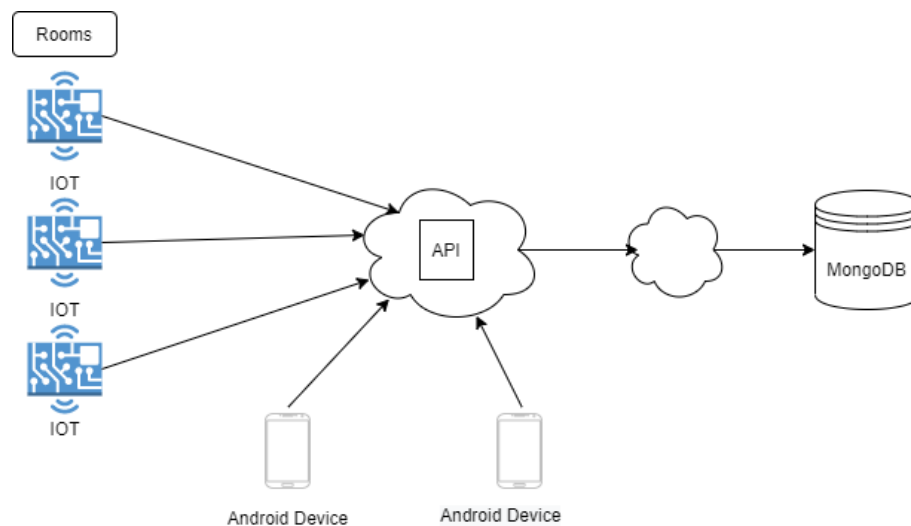


Figure 5.6: Architecture of the System

5.1.3 Security- Roles

In this application, we have three (3) main roles for authentication:

- Student
- Guest
- Admin

Cleaning person is also a role presented in our application, but, when it comes to security, it's not part of it.

The person's passwords are encrypted through JWT (*JSON Web Tokens*) and tokens are valid for one (1) hour.

5.1.4 Integration levels

When it comes to integration levels, the API will receive two (2) different types of clients:

- Embedded systems- residence rooms
- Mobile apps- students and guests

5.1.5 Problem Solution

The main problems that appeared during the evolution of the project tend to be related with connecting the API with the other systems. For example, the API needs to be connected to a C# application so the Arduino can consume its data. This also needs to be connected to the mobile application so the user can control it through its phone. Also, some problems occurred due to the upgraded technologies, such as .NET 7.0.

To solve the above mentioned problems, the group needed to work in all projects at the same time, adding new functions and informations as the program was being developed.

The base of the entire project is a RESTful API that supports all CRUD operations, protected by JWT for authorization. The rest of the project is related to the entire Campus API.

The group didn't see any needs when it came to use SOAP services, since REST services would be faster.

5.1.6 Endpoints

Our API contains six (6) main paths in endpoints:

- api/Student
- api/Admin
- api/Room
- api/Guest
- api/CleaningPerson
- api/cleaningappointment

Depending on the according method, authorizations are personalized in a way that users from determined roles have access to those endpoints.

5.1.7 Code Implementation

The following code is responsible for encrypting the passwords using a private key (*salt*).

```

1  namespace SmartResidenceAPI.Services
2  {
3      /// <summary>
4      /// Hashing class using BCrypt
5      /// </summary>
6      public class Hashing
7      {
8          /// <summary>
9          /// Get random salt from BCrypt to hash password
10         /// </summary>
11         /// <returns></returns>
12         private static string GetRandomSalt()
13         {
14             return BCrypt.Net.BCrypt.GenerateSalt(12);
15         }
16
17         /// <summary>
18         /// Hash password using a generated salt
19         /// </summary>
20         /// <param name="password"></param>
21         /// <returns></returns>
22         public static string HashPassword(string password)
23         {
24             return BCrypt.Net.BCrypt.HashPassword(password, GetRandomSalt());
25         }
26
27         /// <summary>
28         /// Validate password
29         /// </summary>
30         /// <param name="password"></param>
31         /// <param name="correctHash"></param>
32         /// <returns></returns>
33         public static bool ValidatePassword(string password, string correctHash
34         )
35         {
36             return BCrypt.Net.BCrypt.Verify(password, correctHash);
37         }
38     }

```

Listing 5.1: Hashing Code

5.1.8 Conclusion

The application is working correctly, although not everything that was originally planned is implemented.

There is a file (*README.txt*) with the Github repository that have some files, like the URL from Cloud deployment since the URL could change anytime and some instructions how to use this API.

Despite that, the group is planning to implement a Load Balancer ("efficiently distributing incoming network traffic accross a group of backend servers, also known as *server farm* or *server pool*") in the future, since rooms will fetch data in a ms matter in a cyclic way.

Another point to improve revolves around the fact that only RESTful functions were used and the group was planning to dive into the world of Web Sockets, that makes a two-way interactive communication session between the user's browser and the server possible. That way, the client can send messages to a server and receive responses until one of them closes the connection. Since it's bidirectional, it's more complex than HTTP, and also more complicated to implement, reason why this is not (yet) part of our project.

With this retrospection, the group makes a positive balance of the project.

5.1.9 Bibliography

Load Balancing

Web Socket

Chapter 6

Embedded Systems

6.1 Contextualization and Motivation

The goal of this project is to create the hardware based system, to be implemented in the academic residence.

Since the academic residences project was approved, we thought on an idea to develop a smart system to manage the rooms.

The rooms will have a RFID system, a display, a press button, eletronic lock on the doors and a smart light system.

6.2 Architecture of the System

To demonstrate this system, we will show our project in Tinkercad and, later, through a video.

This project integrates a very important component: an Ethernet shield, capable of connecting our Arduino to the Internet through a RJ45 cable. It also has an onboard micro-SD card slot, which can be used to store files for serving over the network.

With the shield connected to the Arduino, we connected the rest of the components:

- a LED to simulate the light system of the room
- a press button on the wall that the cleaning person must press to finish the cleaning service
- a RFID module that allows people to enter the room by reading their card/tag RFID
- an I2C display 16x2 that shows the relevant information
- a motor servo that simulates an automatic lock on the doors
- a photoresistor that detects the brightness of the light
- needed resistors (1 x 220 Ohms)
- needed breadboards (1 x large breadboard)

With these components, we built our system, represented by the circuit on the figures 6.1 and 6.2.

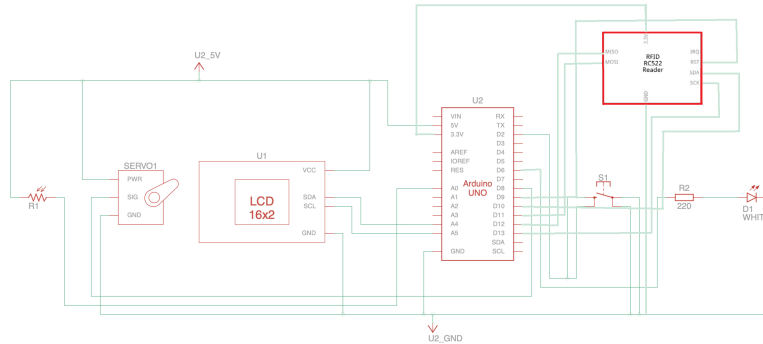


Figure 6.1: Circuit of the residence system

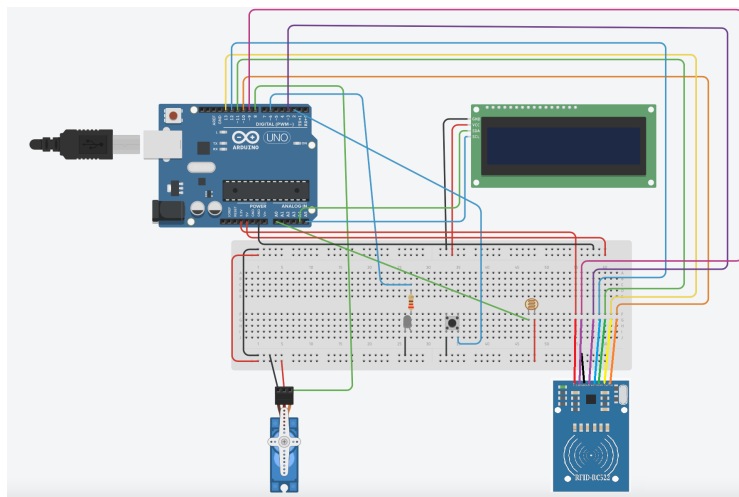


Figure 6.2: Circuit 3D of the residence system

6.2.1 Justification of the measures taken

Originally, we thought about using more than one Arduino Uno, but later we decided to use only one Arduino Uno, in order to sync every data, since, if we had two or more, the other one could need that data and wouldn't have it. In real life, this could be solved by having cables that would connect the components to the main Arduino.

One of the main problems through the development of this project was to connect the system with the Cloud API that is deployed on the Cloud. The idea was to implement another .NET

system that would connect the two, but that would be too complicated. So, instead, we used the Ethernet shield, in order to connect the Arduino to our Cloud API.

6.2.2 Conclusion

The group is planning to implement a gas and humidity sensor.

Despite that, the application is working correctly, although not everything that was originally planned is implemented.

With this retrospection, the group makes a positive balance of the project.

6.2.3 Bibliography

Ethernet Shield

References

Appendix A

Group organisation dossier

A.1 Internal Rules

A.1.1 Scope

The purpose of this document is to inform about the constitution and functioning of Workgroup 14 in the preparation of the project for the Curricular Unit “Applied Project”. This document also informs of the duties and obligations of the members of Workgroup 14.

A.1.2 Workgroup constitution

The workgroup 14 of the Bachelor Degree in Computer Systems Engineering is made up of the following members:

- Bruna Macieira nr. 21139 - Product Owner - Development
- Ricardo Teixeira nr. 20080 - Scrum Master - Development

This workgroup is guided by the Professor Eduardo Peixoto who can participate in group meetings to help achieve the project’s objectives.

A.1.3 Meetings

- Group meetings are scheduled every week at 9:00 pm on Mondays and Thursdays.
- In case of other more important commitments exist, the meetings can be rescheduled.
- Extraordinary meetings can be scheduled when necessary.
- Notices will only be made when there are extraordinary meetings.
- Small minutes are taken whenever a meeting takes place, describing the topics covered.

A.1.4 Internal evaluation

Grades will be assigned at the end of each evaluation moment, and each member describes the performance of the other taking into account attendance, punctuality, participation, interest, communication, respect between colleagues, deadlines, organization, initiative, quality of work and autonomy.

A.2 Timetable

Outubro	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Sab	Dom	Seg	Ter	Qua	Qui	Sex	Sab	Dom	Seg	Ter	Qua	Qui	Sex	Sab	Dom	Seg	Ter	Qua	Qui	Sex	Sab	Dom	Seg	Ter	Qua	Qui	Sex	Sab	Dom	Seg
09.00h																															
10.00h																															
11.00h																															
12.00h																															
13.00h																															
14.00h																															
15.00h																															
16.00h																															
17.00h																															
21.00h																															
Meeting 1																															
Novembro	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
	Ter	Qua	Qui	Sex	Sab	Dom	Seg	Ter	Qua	Qui	Sex	Sab	Dom	Seg	Ter	Qua	Qui	Sex	Sab	Dom	Seg	Ter	Qua	Qui	Sex	Sab	Dom	Seg	Ter	Qua	
09.00h																															
10.00h																															
11.00h																															
12.00h																															
13.00h																															
14.00h																															
15.00h																															
16.00h																															
17.00h																															
21.00h																															
Meeting 2																															
Meeting 3																															
Meeting 4																															
Meeting 5																															
Meeting 6																															
Meeting 7																															
Meeting 8																															
Meeting 9																															

Figure A.1: Timetable 1 - Meetings and Work Planning

Dezembro	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Qui	Sex	Sab	Dom	Seg	Ter	Qua	Qui	Sex	Sab	Dom	Seg	Ter	Qua	Qui	Sex	Sab	Dom	Seg	Ter	Qua	Qui	Sex	Sab	Dom	Seg	Ter	Qua	Qui	Sex	Sab
09.00h																															
10.00h																															
11.00h																															
12.00h																															
13.00h																															
14.00h																															
15.00h																															
16.00h																															
17.00h																															
21.00h																															
Meeting 10																															
Delivery 3																															
Meeting 11																															
Meeting 12																															
Meeting 13																															
Meeting 14																															
Meeting 15																															
Meeting 16																															
Meeting 17																															
Meeting 18																															
Meeting 19																															
Meeting 20																															
Meeting 21																															
Meeting 22																															
Meeting 23																															
Meeting 24																															
Meeting 25																															
Meeting 26																															
Meeting 27																															
Meeting 28																															
Meeting 29																															
Meeting 30																															
Meeting 31																															

Janário	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Dom	Seg	Ter	Qua	Qui	Sex	Sab	Dom	Seg	Ter	Qua	Qui	Sex	Sab	Dom	Seg	Ter	Qua	Qui	Sex	Sab	Dom	Seg	Ter	Qua	Qui	Sex	Sab	Dom	Seg	Ter
09.00h																															
10.00h																															
11.00h																															
12.00h																															
13.00h																															
14.00h																															
15.00h																															
16.00h																															
17.00h																															
21.00h																															
Meeting 10																															
Meeting 11																															
Meeting 12																															
Meeting 13																															
Meeting 14																															
Meeting 15																															
Meeting 16																															
Meeting 17																															
Meeting 18																															
Meeting 19																															
Meeting 20																															
Meeting 21																															
Meeting 22																															
Meeting 23																															
Meeting 24																															
Meeting 25																															
Meeting 26																															
Meeting 27																															
Meeting 28																															
Meeting 29																															
Meeting 30																															
Meeting 31																															

Scope or Subject matter:

Analyse the theme of the project and its context. The overall theme proposed by the teacher was Smart Campus. The theme decided was Smart Residence. Planning of the delivery 1 and 2.

A.3.2 Meeting 2

Date:

03 November 2022

Meeting participants:

Working group and the teacher.

Scope or Subject matter:

Choose the components to implement within the Smart Residence context and discussion of the idea.

A.3.3 Meeting 3

Date:

07 November 2022

Meeting participants:

Working group and the teacher.

Scope or Subject matter:

Explore more possibilities with our theme. Contextualization.

A.3.4 Meeting 4

Date:

10 November 2022

Meeting participants:

Working group.

Scope or Subject matter:

Requirements. Identify the different sensors to use in hardware implementation.

A.3.5 Meeting 5

Date:

14 November 2022

Meeting participants:

Working group.

Scope or Subject matter:

Use cases and Entity-Relation Diagram discussion.

A.3.6 Meeting 6

Date:

17 November 2022

Meeting participants:

Working group.

Scope or Subject matter:

Validate Wireframe and final adjustments on documentation.

A.3.7 Meeting 7

Date:

21 November 2022

Meeting participants:

Working group.

Scope or Subject matter:

State Diagram and Sequence Diagram discussion and validation.

A.3.8 Meeting 8

Date:

24 November 2022

Meeting participants:

Working group.

Scope or Subject matter:

Activities Diagram discussion and validation. Documentation corrections.

A.3.9 Meeting 9

Date:

28 November 2022

Meeting participants:

Working group and teacher.

Scope or Subject matter:

Meeting with teacher. It was decided that we have to correct some aspects on documentation, some descriptions, diagrams, table of contents and references.

A.3.10 Meeting 10

Date:

5 December 2022

Meeting participants:

Working group.

Scope or Subject matter:

Mockups discussion and validation. Documentation corrections.

A.3.11 Meeting 11

Date:

12 December 2022

Meeting participants:

Working group and teacher.

Scope or Subject matter:

Poster decisions and development start discussion and strategy.

A.3.12 Meeting 12

Date:

15 December 2022

Meeting participants:

Working group and teacher.

Scope or Subject matter:

Development work discussion.

A.3.13 Meeting 13

Date:

5 January 2023

Meeting participants:

Working group and teacher.

Scope or Subject matter:

Development work discussion. Solving problems with some electronic components.

A.4 Internal Evaluation

A.4.1 Deliveries 1 and 2

A.4.1.1 Bruna about Ricardo

Ricardo is a great teammate, since he's a hardworking, intelligent and open-minded man. He finds great solutions to solve problems and our minds get in sync really fast when it comes to implement an idea. He's amazing to work with. Evaluation grade (0-20): 19

A.4.1.2 Ricardo about Bruna

Bruna was always interested, concerned and had initiative. She proposed several ideas for implementation in the project and was autonomous in the development of her tasks. We always communicate when necessary even outside of meetings. She was a good co-worker. Evaluation grade (0-20): 19

A.4.2 Delivery 3

A.4.2.1 Bruna about Ricardo

Ricardo couldn't be present much, but he did compensate himself by working remotely. Almost everything went as planned. Evaluation Grade (0:20): 18

A.4.2.2 Ricardo about Bruna

Bruna worked as expected and planned. She was always available to solve any problem or doubt about anything. Evaluation Grade (0:20): 20

A.4.3 Delivery 4

A.4.3.1 Bruna about Ricardo

Ricardo was more present during this stage of the project and everything went as planned. Evaluation Grade (0:20): 19

A.4.3.2 Ricardo about Bruna

Bruna worked as expected and planned. This time we were more united during work. Evaluation Grade (0:20): 19