Ricardo Filipe Rodrigues Ferreira

Licenciado em Ciências da Engenharia Eletrotécnica e de Computadores

Cognitive Notification System for Door Lock

Dissertação para obtenção do Grau de Mestre em

Engenharia Eletrotécnica e de Computadores

Orientador: Rui Manuel Leitão Santos-Tavares, Professor Auxiliar,
Universidade Nova de Lisboa

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2018

Cognitive Notification System for Door Lock

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "novathesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Aos meus pais, Isabel e Manuel, e ao meu irmão Gonçalo.

Acknowledgements

Obrigado Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa. Obrigado por me teres acolhido por mais tempo do que o previsto, por me teres apresentado a algumas das pessoas mais importantes da minha vida e pelas experiências vividas naqueles que foram os melhores anos da minha vida. Até agora.

Agradeço ao Professor Rui Tavares pelo apoio no desenvolvimento deste trabalho, que fez com que pudesse condensar algumas das competências adquiridas ao longo do curso num projeto de grande interesse pessoal.

Obrigado aos meus pais, Isabel e Manuel, pela paciência, pelo apoio incondicional e por tudo o que me ensinaram, e continuam a ensinar, e que me ajudou a tornar na pessoa que hoje sou. Agradeço-vos do fundo do meu coração.

Ao que será sempre o meu mano mais velho, Gonçalo, obrigado pelos sábios conselhos e pela calma que me incutes sempre que tento dar passos maiores que as minhas pernas.

Martim e Maria, obrigado pela ingenuidade e olhos brilhantes que me fazem voltar a ser criança. Que tenham em vocês todos os sonhos do Mundo.

Um obrigado também à restante família, com um especial beijinho para a minha Avó.

Obrigado a todos os que fizeram parte desta caminhada académica com foco especial para o Flávio Jacinto, Francisco Silva, Gabriel Anacleto, Guilherme Góis, João Henriques, Miguel Santos, Pedro Alves, Ricardo Fortuna e Rui Batista. Sem vocês, teria sido muito mais difícil.

Obrigado aos parceiros da 3.5, Andreia Ribeiro, João Duarte, João Simões, Miguel Lopes, Pedro Lopes e Rodrigo Francisco. Sem vocês, teria sido menos divertido.

Obrigado também ao João Pombas e João Veloso por todos os conselhos técnicos ao longo do desenvolvimento deste trabalho. Sem vocês, teria aprendido muito menos.

Obrigado Diogo Santos, João Batista, Jorge Dias, Marta Meirinhos, Micael Fernandes, Pedro Nunes, Rafaela Reis, Raquel Lourenço, Rodrigo Santos e Sara Martins pelas aventuras, experiências e irresponsabilidades.

Não queria deixar de agradecer ao Paulo Gonçalves e ao Thiago Marques pelos bons conselhos e conversas produtivas.

Obrigado ao Agrupamento de Escuteiros 895 de São João da Talha, pelo crescimento, experiências e amizades para a vida.

Obrigado ao Almada Communication Leaders Toastmasters Club pelo espírito de grupo e crescimento sustentado.

Obrigado à OLX, à Spark Agency e à Vodafone, pelas oportunidades de crescimento pessoal e profissional que me deram ao longo do meu percurso académico.

Por último, um agradecimento especial à Lúcia. Obrigado pelo apoio constante ao longo destes últimos anos e por me tornares uma pessoa melhor todos os dias. Esta meta também é tua, muitas mais virão.

Que as próximas aventuras e desafios continuem a ser partilhadas com todos vocês.

Ricardo

Abstract

Most of the time, home automation devices provide peace of mind to their users by helping them in managing their house's heat system, their garden watering or even controlling their home's security. All the home automation devices connect to a network and they gather massive quantities of data that, most of the time, is underappreciated and only used to notify the users about real-time events.

Smart locks provide easy access to buildings and, like other home automation devices, generates a lot of rich data that can be used, and studied, to provide interesting features to their users.

This work proposes a smart lock cognitive notifications system that studies the smart lock activity generated data to provide notifications regarding the user's behavior to the system administrator through a smartphone application.

The developed system comprises a cloud computing component that is responsible for the communications between all the system's modules, data storage and ultimately it runs the cognitive notification system. To generate the needed data for the notifications system a smart lock prototype was also developed. The users can control the smart lock by using a smartphone application that includes several useful features.

The whole system works reliably and can be a good addition to the home automation market by enabling new possible features that will make the users lives more comfortable.

Keywords: Internet of Things, Home Automation, Smart Lock System, Cognitive Notifications System, Cloud Computing, Smartphone Application

Resumo

Na maioria das vezes, os dispositivos de automação residencial ajudam os seus utilizadores a atingir uma maior paz de espirito através do controlo automático do seu sistema de aquecimento, da rega do seu jardim ou no controlo do sistema de segurança. Todos os dispositivos de automação residencial podem estar ligados em rede e com isso juntar uma quantidade assinalável de dados que, na maioria das vezes, é subaproveitada e acaba apenas por ser utilizada para notificações de eventos em tempo real.

As fechaduras inteligentes fornecem acesso fácil a edifícios e, tal como outros dispositivos de automação residencial, geram grandes quantidades de dados que podem ser usadas, e estudadas, para criar novas e interessantes funcionalidades para os seus utilizadores.

Este trabalho propõe um sistema cognitivo de notificações que estuda os dados gerados pela atividade de uma fechadura inteligente e que providencia notificações ao administrador do sistema, através de uma aplicação móvel, baseadas nos comportamentos dos utilizadores.

O sistema desenvolvido compreende uma componente de computação na nuvem que é responsável por todas as comunicações entre os módulos do sistema, pelo armazenamento de dados e é onde o algoritmo do sistema cognitivo de notificações efetua as suas rotinas. Com o objetivo de gerar os dados necessários às notificações, foi desenvolvido um protótipo de uma fechadura inteligente. Os utilizadores deste sistema podem controlar a fechadura inteligente através de uma aplicação móvel desenvolvida para esse efeito e que inclui várias funcionalidades úteis.

O sistema funciona de modo sólido e pode ser uma boa adição ao mercado da automação residencial podendo habilitar outros sistemas de novas funcionalidades que tornarão as vidas dos seus utilizadores mais confortáveis.

Palavras-chave: Internet das Coisas, Automação Residencial, Sistema de Fechadura Inteligente, Sistema de Notificações Cognitivas, Computação na Nuvem, Aplicação Móvel

Contents

List of Figures

List of Tables

Acronyms

AES	Advanced Encryption Standard.
AWS	Amazon Web Services.
BLE	Bluetooth Low Energy.
DC	Direct Current.
GPIO	General Purpose Input Output.
HTTP	Hyper Text Transfer Protocol.
HTTPS	Hyper Text Transfer Protocol Secure.
IDE	Integrated Development Environment.
IFTTT	If This Then That.
IoT	Internet of Things.
LED	Light-Emitting Diode.
M2M	Machine To Machine.
NO	Normally Open.
SQL	Structured Query Language.
UNB	Ultra Narrow Band.
URL	Uniform Resource Locator.
USB	Universal Serial Bus.
UWB	Ultra-Wideband.

WLAN	Wireless Area Local Network.
WPAN	Wireless Personal Area Network.
XML	Extensible Markup Language.

Chapter 1

Introduction

Technology is evolving every day. In most cases, technology improves the lives of others on a daily basis, making them easier or simply more productive.

Home automation is one branch of technology that has the aim to boost people's lives just by automating processes that people have to manage each day in their homes. Simple processes like, opening or closing your blinds, managing the temperature of your heating system or controlling your home's illumination can be automated according to users preferences and needs. This is the kind of technology that gives the users more time to put into really important things, making the users day more efficient and productive.

Home automation is divided into some areas. One of the most relevant is security, where we can find devices like smart locks, surveillance cameras, and smart alarm systems. These devices are very relevant because they can give the users access to their homes, but since they're smart devices that can be connected to the Web, they should be developed and designed properly to give real security to the product users.

1.1 Motivation

The Internet of Things (IoT) concept isn't just the future, it's already the present. Gartner forecasts said that in 2017, 8.4 billion things would be connected to the internet, which means a 31% expansion from the year before that. Their forecasts also predict that by 2020, those devices will reach the number of 20.4 billion, with an expected value of $2 trillion between devices/endpoints and services, [1].

In a near future, everything around us will be collecting, sending and receiving data for a lot of different purposes and goals.

Contributing to that massive growth and expansion is the home automation market. According to Mike Krell, who summed up the 2015 State of the Smart Home from iControl

Networks, users that are searching for home automation systems, are mainly looking for systems that don't require any interaction, make their home more safe, energy efficient and that automate themselves as persons, [2][3].

The need for the security feeling is not new to humans, but the devices that provide it have made a long way to the ones in the present. The oldest lock system ever found was made of wood and supposedly belonged to the kingdom of Assyria. Egyptians and Romans upgraded it. By the 18th century, Joseph Bramah developed a system similar to today's lock where the key lifted different heights in order to a corresponding pattern on the inside. In 1975, due to security problems in hotels, Tor Sornes developed and patented the first electronic keycard lock. Since then, the world has seen a massive evolution in the door lock systems, [4].

Smart locks will be definitely a big part of the Smart Home Market. Michael Wolf said, in the beginning of 2014, that were nearly 1 million homes in North America using some form of smart lock, while the predictions are for this market to hit $3.6 billion worldwide by 2019, [5].

Supporting all these innovations and emerging markets is the rates of adoption and usage of smartphones. Pew Research Center made a study which concludes that 95% of Americans own a cell phone of some kind, with 77% of these being smartphones, [6].This will surely have a great influence in the home automation market since most of the already available devices are controlled through smartphone applications making home automation solutions even more attractive to costumers.

With the IoT/Home Automation growth and expansion and with people looking for devices that can grant them more safety and automation in their lives, it's very important that home automation devices, especially smart locks, can learn from their users habits providing them more value, automation, and functionality each day. This kind of very smart devices will make people more focused on the things that really matter in their lives while owning home automation devices that seem to be built for their own needs.

1.2 Objectives

This work aims to add a contribution to the home automation industry, especially to the smart lock market.

This work goal is to develop a smart lock cognitive notification system that can predict the smart lock users behaviors and give alerts to the system administrator in order for him to know if their kids reached home safely, if they already should be at home or if any different activity from the usual is happening. To properly develop the notification system an A to Z smart lock solution should be developed. That system should contain a simple smart lock prototype that is controlled through a smartphone application. A cloud platform must exist to control the system module's interactions and to store all the generated data.

In order to contribute properly, a market approach should be made in order to assess the smart lock market needs and how can we develop a new solution that can complement the ones already on sale.

To develop a new solution it's necessary to study the most recent technologies that can be used in our solution, their vantages, and disadvantages in order to reach the final goal.

The developed solution must be tested in order to correct the problems and bugs that may appear, this will make the final prototype more reliable.

1.3 Document overview

This document is structured in the following way:

- **Chapter 2** reveals the State of the Art of IoT and Home Automation market and technology. The chapter is focused on doing an assessment of the smart locks market in order to evaluate the features and the operation of several smart locks that have already hit the shelves. The focus is also on the technology present in the IoT and Home Automation devices in the form of an overview of several technologies;
- **Chapter 3** describes succinctly a proposed solution, it's functionalities and how the system should behave;
- **Chapter 4** provides an inside view of how the system was implemented and the technologies that were used to make it happen;
- **Chapter 5** is focused on this work's results and an approach to future developments.

Chapter 2

State of the Art

Home automation market is booming. At the same time, new products are launched, with a growing frequency, into the market with the aim to suppress all the user's needs, or even to create new ones.

In this chapter, it's presented a smart lock market approach of two of the most recent smart lock solutions that are supporting the market growth. The goal is to assess what has already been done and which value is this work providing to the market. The presented solutions define the benchmark for both the European and North American smart lock markets because of their high security standards, quantity and quality of features, such as auto-unlock and remote unlocking, and also by showing a great user experience through their smartphone applications.

2.1 Smart lock market overview

The presented solutions are some of the best in the developing smart lock market. Despite the fact that all of them provide their user's great sets of useful features, none of them provides a notification system that is based on previously acquired knowledge like the one that is proposed by this work.

2.1.1 Nuki

Nuki Home Solutions is an Austrian company who aims to develop products that turn your home into a smart one. Their main product is Nuki Smart Lock, specially developed with the European market in mind. The Nuki Smart lock is fitted on the inside of your door, in the top of the key and it works by rotating it, [7].

Nuki provided their Smart Lock with Bluetooth Low Energy (BLE) in order to connect the smart lock to the smartphone application In order to use all the features available,

Nuki Smart Lock must be connected to the web with Nuki Bridge that provides a Wi-Fi connection to the smart lock. Nuki Bridge is sold separately.

To use the smart lock, the user should have a smartphone with the Nuki smartphone application installed. There, the user can manage and configure his Nuki Smart Lock.

Nuki Smart Lock operations are powered by 4 AA batteries with an estimated battery life of 6 months, calculated with 8 locking processes per day. When the smart lock has only 20% battery left, the user receives a notification from the Nuki smartphone application suggesting the batteries replacement. If the smart lock batteries die, the user can continue to access his house using his physical keys. There's another scenario to consider, the one where the user smartphone battery dies. In that case, there are several options to unlock the door. The user can use his physical keys, ask another authorized user to unlock it for him or use Nuki Fob, a small Bluetooth device that can connect to Nuki Smart Lock.

Security is one of the most important specifications for home automation products. Nuki provided their smart lock with BLE to communicate with the Nuki App, but instead of relying only on Bluetooth protocol security, they added an additional layer of security on top of it with end-to-end encryption, which means that only encrypted messages are sent and only the Nuki smartphone application and Nuki Smart lock know the key to decipher the message and perform the correspondent action as exemplified in figure 2.1.

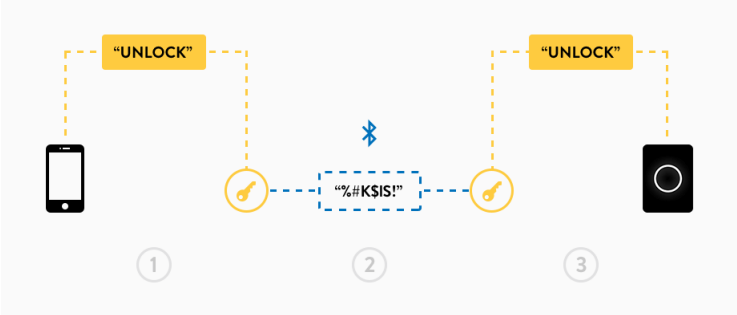


Figure 2.1: Example of a message sent through Bluetooth from Nuki smartphone application to Nuki Smart lock, taken from [8].

Nuki guarantees that the same thing happens if you use Wi-Fi technology and Nuki Bridge to perform actions with your Nuki Smart Lock, with no data being saved on the Nuki servers, this action is exemplified in figure 2.2.

This smart lock can be improved by using another Nuki devices:

- **Nuki Connect** - This device adds web connection to the Nuki Smart Lock. It enables remote features like locking and unlocking the door and checking the state of the lock despite the user's location;

- **Nuki Fob** - Small Bluetooth device that connects to the Nuki Smart Lock in order to lock/unlock it. Helps the user to access is house without a smartphone and physical keys;

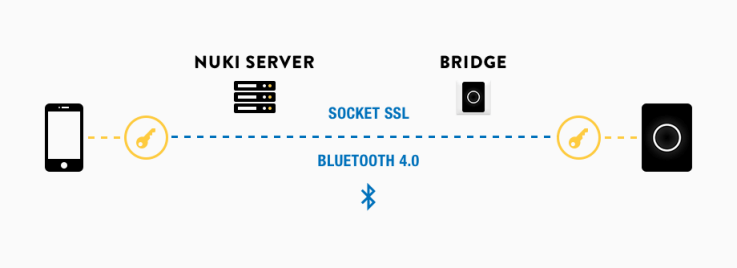


Figure 2.2: Example of a message sent through the internet from Nuki smartphone application to Nuki Smart lock, taken from [8].

- **Nuki Box** - This device aims to help the Nuki Smart Lock users that live in buildings, allowing them to control remotely the building's door through the Nuki App.

This smart lock has a very wide set of features available:

- **Locking with the Smart Lock** - With Nuki Smart Lock the user can lock or unlock the door by pressing the smart lock button. Alternatively, he can execute the same functions using the smart lock knob as if he would turn the mechanical key;
- **Locking with the Nuki App** - Using the Nuki App, the user can lock and unlock his smart lock with his smartphone, making it easier to enter and leave his house. With Nuki Bridge it's possible to perform remote locking/unlocking;
- **Lock 'n' Go** - This feature gives the user more comfort. With the push of a button the user can leave his house with the certainty that his door will lock automatically for him;
- **Auto Unlock** - This feature allows the user to unlock the door only by getting near it. It uses Global Positioning System, Bluetooth, and Geofences to help locate the user and sense when he's near the door to connect the Nuki Smart Lock with the Nuki smartphone application and proceed with the unlock;
- **Integration** - Nuki Smart Lock has integration with Amazon's Echo and Alexa and Google Home and Assistant, providing some voice commands that can control the smart lock. The smart lock has also IFTTT (If This Then That) integration, a free way to put apps and devices talking to each other, [9].
- **Installation** - Nuki Smart Lock is very easy to install since it doesn't require any change to the regular lock in the door, which can be very useful for the average user who just wants to have a ready to use product on his hands, or in this case, on his door. Nuki Smart Lock is suitable for doors with handle or knob on the outside. The user can configure the smart lock to unlock the door or even to pull the latch, which means that it can open the door for him;

- **Nuki smartphone app** - With the Nuki App, the administrator can control and manage every aspect of his Nuki Smart Lock. Using the application it's possible to lock and unlock it, with Nuki Connect it can be done remotely. The application can also be used to check the lock activity log, see and manage the authorized users or invite new ones. The administrator is also able to invite temporary users by configuring a certain date and hour when that invite will be valid. An invited user receives an invitation code by text message or email and has to insert it into the Nuki smartphone application to gain access to the Smart Lock.

2.1.2 August

August, Inc is a company based in San Francisco, with focus on making locks more smart and helpful. Their high-end and main product is August Smart Lock Pro, which can connect to other secondary equipment's that improve its mission of smartly unlocking doors. This smart lock should be installed on the inside of the door and is compatible with most deadbolt systems used in the United States of America houses, [10].

August equipped their main smart lock with three wireless communication protocols: Bluetooth, Wi-Fi and Z-Wave Plus. The lock has integration with Amazon's Echo and Alexa, Google Home and Assistant, and Apple's HomeKit and Siri, which enables a smart home where all devices are connected to a centralized hub where the user is able to control everything. The August Smart Lock Pro has also If This Then That (IFTTT) integration.

August Smart Lock Pro is powered by 4 AA batteries with an estimated battery life of 3 months. The user will receive two different warnings when his smart lock battery is low, he will receive a smartphone application (app) notification on his smartphone, and the smart lock will display a red flashing light. Although, if the smart lock batteries or even the user's smartphone battery dies, the door lock it's still usable with the physical key or even by logging into the user's August account in a friend's or neighbor phone.

The communication between the smartphone and the smart lock are always established via BLE. The packets exchanged between the two devices are used for the lock to authenticate the phone, grant or revoke user access and to update the smartphone application about who has locked/unlocked the lock. These packets are encrypted using Advanced Encryption Standard (AES) encryption, that is a symmetric encryption algorithm where both parties must send ciphertexts encrypted with the firmware key. But while the owner, or main user, has a unique key per session, where the smartphone has an offline key and only communicates with the lock to perform different actions, the guest user, or guest smartphone, does not have access to an offline key, so in order to interact with the lock, he has to communicate with the server which works as a middleman between the guest smartphone and the smart lock. Without exploring more of the security protocols present in the August devices and app, it's important to notice that in Fuller, Jenkins and Tjølsen security review, no major vulnerabilities were found in the device, [11].

Regarding different user's needs, August has very good additions to their smart lock. The user can update his smart lock with the following devices:

- **August Smart Keypad** - Exterior numerical keypad which enables the door unlocking action just by inserting a numerical code;
- **August Doorbell Cam Pro** - It aims to replace the user's existent doorbell. It has a camera built-in that connects to the application helping the user know who is at his front door. At the same time, he can see and speak with visitors using his phone;
- **August Connect** - Provides web connection to the smart lock by connecting to the user's house Wi-Fi. Enables certain smart lock features that are only available through the Wi-Fi connection;
- **DoorSense** - This technology uses a magnetic field to determine if your door is closed or open. This device, that is placed near your smart lock, enhances the user experience.

August also incorporated several features in order to attend all their user's necessities:

- **Locking with the Smart Lock** - With August Smart Lock Pro, the user can unlock his door, from the inside of his house, just by rotating the smart lock just like using the door's original deadlock;
- **Locking with the August Home App** - Using the August Home App, the user can lock and unlock his smart lock with his smartphone just by pressing a button. Remote locking/unlocking is available with August Connect device;
- **Auto-Lock** - With DoorSense sensor, the user can configure his smart lock to lock right after it's closed, or after a configurable number of minutes. This feature also assures the re-lock of the door when the user unlocks it but don't open it for some reason;
- **Auto-Unlock** - With Auto-Unlock feature, the smart lock knows and senses when the user and his smartphone are approaching the door and unlocks it without the user reaching his phone. The smart lock has two states, the Home Mode and Away Mode. When the lock is in Home Mode he is not looking for devices to get near, which prevents undesirable locking and unlocking actions. When the user leaves the neighborhood of his house, the lock enters in Away Mode, which means that the lock is now waiting for the user's smartphone to enter in the house area to start the search in order to establish a connection with it and to execute the door unlocking for him;
- **Remote monitoring and control** - With the August Connect, the user is able to monitor and to control his door and smart lock states from remote locations;

- **Integration** - August Smart Lock Pro has integration with several devices and services such as Amazon's Echo and Alexa, Google Home and Assistant, and Apple's HomeKit and Siri, which enables voice control over the smart lock. IFTTT integration enables the smart lock to connect with other smart devices, like smart Light-Emitting Diode (LED) lights, and provides the creation of interactions and routines between those smart devices;

- **Installation** - Installing the August Smart Lock Pro is easy. The user can do it just by removing the inside of the door's deadbolt and by fixing the smart lock there. The use of the physical keys remains available since there aren't any changes on the outside lock hardware;

- **August Home smartphone app** - The August smartphone application is the main connection with their smart lock. There are two types of user, the Owner, and the Guest. The Owner can control and monitor everything that happens with the smart lock, locking and unlocking it (remotely or not), the lock and door state (locked, unlocked, open, closed, left ajar), activate/deactivate features, activity logs and invite/block guests. The Guest user can control the lock only by invite, it means that he has to be invited in order to use the lock. This invite can be a 24/7 one, or the Owner can define the day and hours that the invite will be valid for that specific guest. The smartphone application has also a smart alert function that sends notifications to the user reporting smart lock activity and smart lock and door states. Those notifications can be created by the user, but there's no option to build the type notification as the one that is developed in this work.

2.2 Technology review

Internet of Things (IoT) and Home Automation devices rely on several technologies to work properly and seamlessly. In order to build a smart lock prototype, there is the need to review some of the most common technologies used in that kind of products, such as wireless communications, cloud platforms, sensors/actuators and . This review intends to be a general one, in order to select the most suitable and cheap technologies to apply in the smart lock prototype.

2.2.1 Wireless communication protocols

Wireless technologies have become very popular since they can reduce the costs of implementing automation in homes or other buildings by reducing the amount of cabling necessary to install devices like sensors. This kind of technologies also provides easiness associating smartphones with the automation system, as long as it can connect to the automation network, [12].

In Home Automation there are several wireless communication protocols that can be used. The choice always depends on the devices, sensors, and actuators that are going to be connected and also on the features that the system is supposed to provide.

Considering the fact that this protocols of communication operate on an open medium, security has to be one of the concern when creating home automation solutions that make use of these protocols. Especially if those solutions are security-focused, like a door lock or an alarm system, since attackers can take over unsecured systems without even entering the building. As an additional issue, security features are limited by the requirement of low power consumption on the devices, meaning that a good relation security/power consumption has to be reached in order to reach the device/system requirements, such as price, power consumption among others.

This study [13] provides a general overview and comparison of four short-range wireless communication protocols with low power consumption, Wi-Fi, Bluetooth, Ultra-Wideband (UWB) and ZigBee.

While Wi-Fi and UWB provide a high data rate, making them suitable for systems that need to send/receive a considerable amount of data, Bluetooth and ZigBee are intended for lower data rates. Wi-Fi is also designed to Wireless Area Local Network (WLAN), making it easier to provide a connection in a maximum $100m$ radius, while Bluetooth, UWB, and ZigBee have been developed with Wireless Personal Area Network (WPAN) in mind providing a connection in a maximum $10m$ radius. In some applications, Zigbee can also reach 100m range.

This study also approaches the different frequency bands used by the different wireless communication protocols. Wi-Fi uses $2.4GHz$ or $5GHz$, Bluetooth also uses the $2.4GHz$ frequency band, while UWB can use any frequency between $3.1GHz$ and $10.6GHz$, Zigbee can use frequency band between $868MHz$ and $915MHz$ or it can use the same frequency as Bluetooth, $2.4GHz$. This is an important parameter to evaluate since that radio channels in the $2.4GHz$ band are unlicensed in most countries and is known as the industrial, scientific and medical (ISM) band, however, it is excessively crowded. While high data rate applications have no other alternative, home automation applications can work by using lower frequencies, with the advantage of better wave propagation with the same amount of power spent, [12] [13] .

With Wi-Fi, Bluetooth and Zigbee sharing the same frequency band it's important to notice that Bluetooth and UWB use adaptative frequency hopping in order to avoid channel collision with other protocols, while Zigbee and Wi-Fi use adaptative frequency selection that changes the frequency to another one where the protocol can operate.

All the four protocols have data encryption and authentication mechanisms as shown in table 2.1.

Table 2.1: Comparison of different wireless technologies, adapted from [13]

Specification	Communication Protocol							
	Wi-Fi		Bluetooth		Zigbee		UWB	
Nominal Range [m]	100		10		10-100		10	
Frequency Band [GHz]	2.4 or 5		2.4		0.868/0.915 or 2.4		3.1 - 10.6	
Maximum Signal Rate [Mb/s]	54		1		0.250		110	
Normalized Energy Consumption [mJ/Mb]	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx
	<50	<50	100 - 150	100 - 150	300 - 350	250 - 300	<50	<50
Power Consumption [mW]	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx
	600 - 800	600 - 800	<200	<200	<200	<200	600 - 800	600 - 800
Encryption	RC4 Stream cypher, AES block cypher		E0 Stream cypher		AES block cypher		AES block cypher	
Authentication	WPA2		Shared Secret		CBC-MAC		CBC-MAC	

Shahzad and Bengt, compared ZigBee, Bluetooth Low Energy (BLE), and Wi-Fi for in-sensor processing versus raw data transmission. This study aims to check which one of those technologies is the most energy efficient one by checking the power consumption under different scenarios. The study concludes that for data load under 500 bytes ZigBee it's the technology that spends the less amount of energy, on the other hand, Wi-Fi is best suited for data loads over 800kB while BLE it's the best solution, of the three, for data loads between 500 bytes and 800kB, as is case for typical data-intensive monitoring applications, [14].

Sigfox was developed with the goal to offer a low-cost solution to connect all machines and objects to the internet. It uses a patented Ultra Narrow Band (UNB) wireless communication technology and a cellular data transfer network optimized for Machine-to-Machine (M2M) and IoT applications like building automation, smart cities, asset tracking and others. Sigfox uses free frequency radio bands (license-free) to transmit data between a rate of 1b/s and 1kb/s, it can make transmissions in a 40km radius in open space and it only needs three base stations to cover a city with a million habitants. Sigfox company say they have the best price cellular communication service to M2M and IoT applications, [15]

This work aims to develop a simple smart lock prototype that can be controlled by a smartphone, at the same time there is the need to develop an online database in order to study the data generated by the smart lock usage and all the web communication to assure the integration of all the different modules. With this in mind, the most suitable technology to support the communications of the project is Wi-Fi. Wi-Fi communication protocol provides a reliable way to communicate with the device, that can easily connect to the web in order to receive requests from the cloud. Wi-Fi also enables the feature of locking and unlocking the smart lock from anywhere in the world if a web connection is available.

2.2.2 Cloud platforms

In order to connect all the work modules, and also to store and study the data generated by the smart lock usage, there is the need to use a cloud solution for this work. The selected cloud platform will be the heart of all the system by managing all the smartphone application requests and all the interactions with the smart lock hardware itself.

To select the most suitable solution available we will need to evaluate the different specifications of each one, such as the offered storage capacity, how many GET and PUT data movements are available, and the overall price for using the service. It's also necessary to consider the overall reach of the platform in order to not limit future works after this one.

Despite the high number of cloud solutions available, it's necessary to consider the presented metrics in order to choose the right solution for this work.

- **Amazon Web Services (AWS):**

  This Amazon service has specific products for IoT solutions with a big focus on IoT device management and data security. It has also an integrated data analytics service to help anyone to gather intelligence from their devices generated data.

  Every different product from Amazon has it's own price, depending on which solutions the work will require. AWS products and services are available on a worldwide scale, [16].

- **EasyIoT:**

  The EasyIoT cloud platform has the goal to provide a secure, reliable and cheap platform for IoT, mainly focused in Do It Yourself (DIY) products.

  EasyIoT seems simple to use for the average user, but can't provide as many services and functions as the big names on the market. It's free to use unless the user wants to scale up its solution. It's ideal for single users trying to deploy simple home automation in their own homes, [17].

- **Google Cloud:**

  Google Cloud provides a worldwide cloud solution. Just like AWS, Google Cloud provides a wide range of products and services, including IoT solutions, such as Cloud IoT Core that provides secure device connection and management. Other products to compute and study data are also available.

  Every different product from Google Cloud has it's own price, depending on which solutions the work will require, [18].

- **IBM Cloud:**

  Just like the other big names in the cloud platform industry, IBM has lots of services and products available on a worldwide scale.

Internet of Things Platform, from IBM, helps to deploy and connect IoT solutions. Also, it has several data analysis functions where the user can define its own rules and trigger alerts and other functions.

Every different product from IBM Cloud has it's own price, depending on which solutions the work will require, [19].

- **Microsoft Azure:**

  Microsoft couldn't step away from the cloud industry. It has a lot of services and products available including IoT focused solutions. Just like the competitors, Microsoft has device management and data analytics build specifically to empower IoT solutions and devices.

  Every different product from Microsoft Azure has it's own price, depending on which solutions the work will require, [20].

2.2.3 Sensors and actuators

The door lock prototype that will be developed, will also rely on two sensors and one actuator. This work will need a sensing technology that allows the system to understand if the door is open or closed to avoid unnecessary lock/unlock actions, another sensor to assess movement/presence at the door to add a security layer in the unlock actions and an actuator that should perform the unlock action *per se*.

- **Door position or motion detection technologies**

  The door position or motion sensors will assess the state of the door, if it is opened or closed, in order to avoid unnecessary unlock/lock actions, such as locking the door while the door is open. It also develops an important role in registering new entry activities to the database whenever the door is unlocked and opened by a user.

  There are several sensing technologies that can be used to monitor objects position. To know a door position the cheapest and with more ease of assembling in a door are the accelerometer. The ADXL335 is a low power complete 3-axis accelerometer, it's very small and has an operating voltage between 1.8 V and 3.6 V, [21]. Although the accelerometer has great qualities, it would be probably difficult to understand what's the door positioning by measuring the acceleration of the door movements.

  Magnetic sensors are a very widely used technology in doors and windows. This type of sensors usually connects to the building alarm system to prevent burglars to enter. Two different technologies of magnetic sensors have been researched, the hall effect sensor and the reed switch sensor. There's no big difference between them since they are both activated by an external magnetic influence. Despite that fact, reed switch sensor is known for being more stronger and robust than hall effect sensors, [22]. Also, there is some reed switch sensor that has been made with

door and windows in mind [23] and that is already widely used in performing the functions that this system requires.

- **Presence detection technologies**

  To add a security layer to the system this work will have a presence detection method. There are several presence detection technologies available, this review will highlight three of them.

  - **Ultrasonic sensors** - Ultrasonic sensors are used to detect people and objects. It analyzes its surroundings by sending ultrasonic waves and to study how they are reflected back, this allows the sensor to detect motion. This type of sensor is usually used in home alarm systems. The main disadvantage of this type of sensing technology is that background noise and motion can sometimes trigger the sensor and perform a false alarm signal, [24].
  - **Microwave sensors** - Microwave sensors uses electromagnetic radiation. Its functioning is similar to the ultrasonic sensor, it emits electromagnetic waves which are reflected back to the receiver. If an object or person motion is detected the receiver will identify changes in the electromagnetic waves. Microwave detectors can go through walls and holes. Because of this, they can cover a larger area of a home, [25].
  - **Pyroelectric Infrared (PIR) sensors** - PIR sensors are used to detect human bodies through infrared radiation. They are built to detect the infrared wavelength that human bodies emit. PIR sensors are low-cost, low-power and provide a reliable indication of people presence [26]. This type of sensors doesn't require any device or signal from a detecting object, unlike ultrasonic and microwave sensors. Processing data from PIR sensors is much easier than from other motion sensing technologies, [27].

- **Door lock rotation**

  The simplest method to simulate the door lock, or key lock, rotation, similar to the smart lock systems presented in section 2.1, is to attach a small DC servomotor to the chosen microcontroller.

  This work's focus is to create a cognitive notification system, so the chosen servomotor will only simulate the door lock rotation since the prototype will not be assembled in a real door.

  The DC servomotor has the function to rotate the key or other structure that shall be developed, in order to lock and unlock the door.

2.2.4 Microcontrollers with wireless communications

To control and connect the sensors and actuator and also to receive/send data between the door lock prototype and the cloud platform this work needs a microcontroller that can give us all that.

To select the microcontroller from the ones available on the market, there is the need to define the specifications that are crucial to the development of this work, such as, number of General Purpose Input Output (GPIO) pins must be enough to support the magnetic sensor and the DC servomotor, it has to have enough voltage to supply the DC servomotor and, at the same time, it has to include, at least, the capability to connecting to other devices through Bluetooth, Wi-Fi or other wireless communication protocol. The pricing and ease of buying it are also relevant to the selection of the microcontroller.

Considering all the mentioned aspects, this section presents a comparison between some microcontrollers available in the market.

- **ESP8266 Development Board (NodeMCU V1.0)**

  ESP8266 it's a low power microcontroller focused on Wi-Fi connection. It has an operating voltage between 3.0V and 3.6V, with an operating current average value of 80mA. It has 10 GPIO pins to provide connection to other peripherals. It also has several different working modes in order to minimize the power consumption. In the deep sleep mode it has a typical power consumption of 10μA. This microcontroller can be coded through Arduino IDE and also in Lua programming language, [28] [29].

- **ESP32**

  ESP32 is a highly-integrated solution for Wi-Fi and Bluetooth IoT applications, since it has a Wi-Fi module and a Bluetooth module. It has an operation voltage between 2.3V and 3.6V. ESP32 includes 34 GPIO pins to connect the microcontroller to external sensors and actuators. Just like ESP8266 ESP12 E, it can provide a Deep Sleep mode in order to consume less power. In the deep sleep mode it has a typical power consumption of 10μA. This microcontroller can be coded through Arduino IDE and also in Lua programming language, [30].

- **Bean LightBlue**

  The Bean LightBlue has embedded Bluetooth communication module, an accelerometer, a temperature sensor and an RGB LED. It works with a coin cell battery and has an operating voltage between 2.0V and 3.6V. Bean Lightblue has 8 GPIO pins to connect the microcontroller to external sensors and actuators. This microcontroller can be coded through Arduino IDE, [31].

Chapter 3

System Modeling

To achieve this work's objectives mentioned in 1.2 there is a need to define the solution that will lead there.

The developed system will be divided into three main modules as shown in figure 3.1:

- **Cloud platform**

  The cloud platform is the operations center of the system. It's job is to answer and manage the user's smartphone requests, to interact with the smart lock hardware and to study the activity generated data in order to deliver the main goal of this work, the cognitive smartphone application notifications.

- **Smartphone application**

  The smartphone application communicates with the cloud platform through the web by sending and receiving requests from the cloud. Either is a request for the user to log in or to unlock the door, every user interaction with the system is made through the smartphone application requests.

- **Smart lock hardware**

  Through a microcontroller, a magnetic sensor, a small DC servomotor, and a presence sensor the hardware will represent the smart lock itself. The microcontroller provides web connection, through Wi-Fi technology, that enables the connection of all the hardware with the cloud platform. This allows the three modules to be connected between them through the web, making the cloud platform the center of all system's operations.

In order to understand all the system features and to develop the system itself, it's important to start with the system modeling.

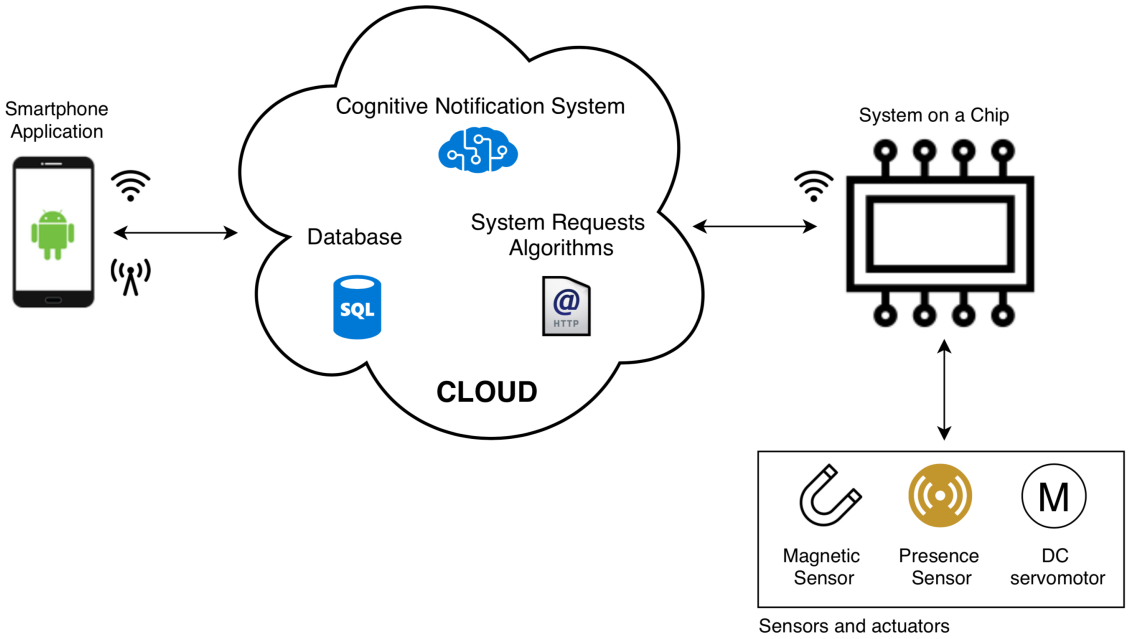


Figure 3.1: System prototype diagram

Section 3.1 is focused onto individually explaining the role of all the major modules of the system and to properly understand the importance of each one in the whole system.

Section 3.2 presents a system model that describes the whole system operation and behaviour.

Section 3.3 makes an approach to all the different system functions, in order to understand what is happening behind each system function.

Section 3.4 aims to describe the data model that is necessary to implement so that the system works flawlessly and, at the same time, ensuring that every function requests are met.

3.1 Use cases

This section aims to describe the different functionalities, roles and use cases that define each one of the system's main modules. Figure 3.2 represents the system use case diagram, where each one of the system's main modules is represented by an Actor that is linked to the use cases that each Actor performs. This diagram is focused on each module main use cases. A deeper approach to each one is made in the respective module's section.

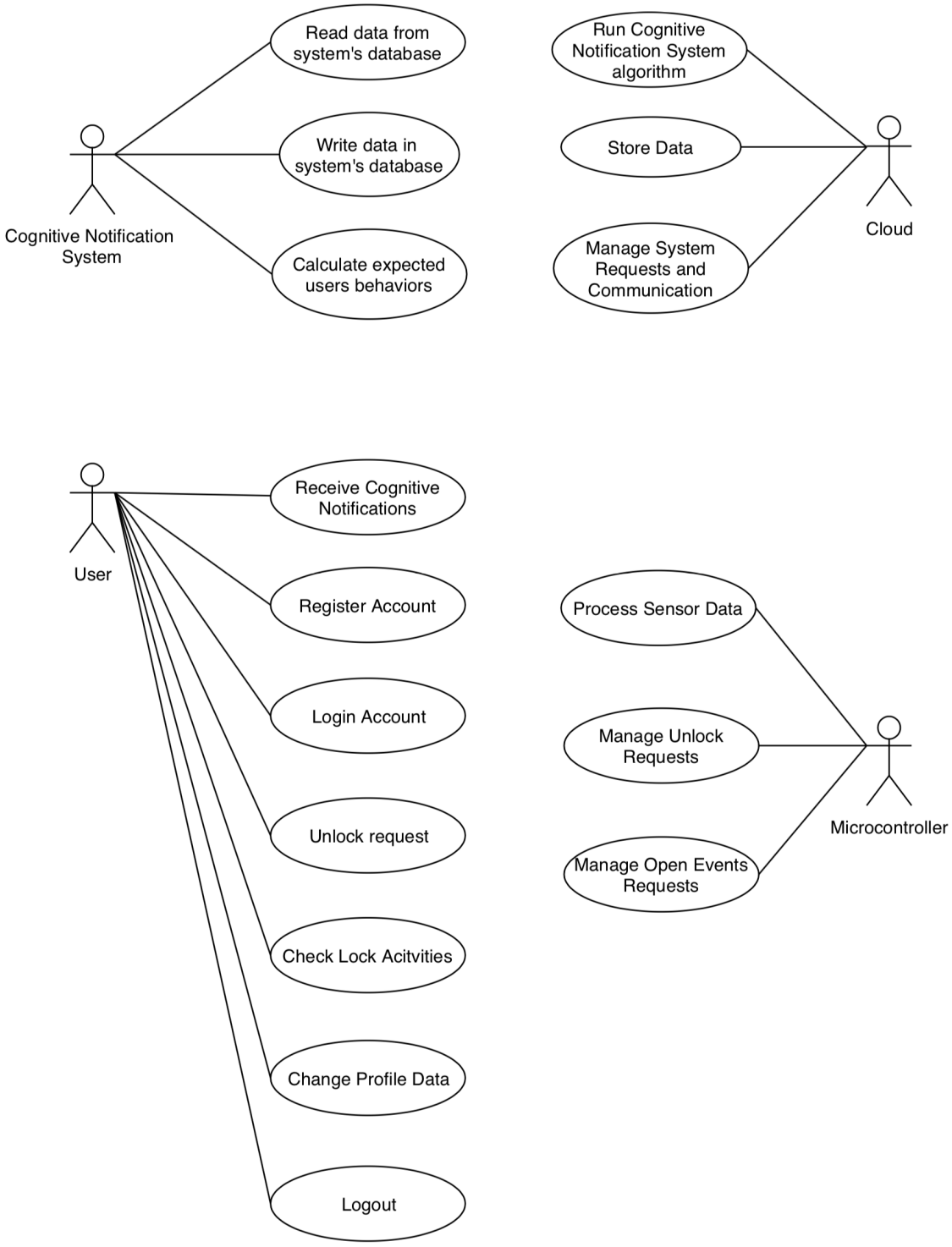


Figure 3.2: Main use cases for each system module

3.1.1 Cognitive notifications

Despite the fact that the notifications algorithm runs inside the dedicated server at the Amazon Web Services (AWS) cloud platform, it's the most important part of all the work, making it fair for them to be considered a main system module.

The smartphone application cognitive notifications goal is to help the system administrator to monitor the smart lock usage and its users. This could be an important feature either for a company to control its employee's arrivals at the office, for parents who want to be warned when something unusual happens with their kids or even if the system administrator wants to be warned when the maid has not arrived in time at his house.

They are called cognitive notifications because they are based on previously acquired knowledge that is stored in the system's database. Since every user has its own habits and routines, every time they use the smartphone application to access the building through the smart lock an activity log is generated for that specific user that helps to generate the cognitive notifications. Despite both the system administrator and regular users can use the smart lock, they play two different roles on the notification system:

- **System administrator:** can use the all the smartphone application and smart lock functions and it's the only one who receives the notifications that are generated by the smart lock usage;
- **Regular user:** uses the smart lock just like the system administrator but is unable to receive notifications that may warn about other users behaviours and routines. Each user activity helps to build the knowledge needed to generate the notifications.

A notification is sent to the system administrator when a user has a different behaviour than the previous ones that he had, so if a specific user unlocks the smart lock always around the same time, but there's one day that he doesn't, a notification is fired to the system administrator letting him know that that specific user is late.

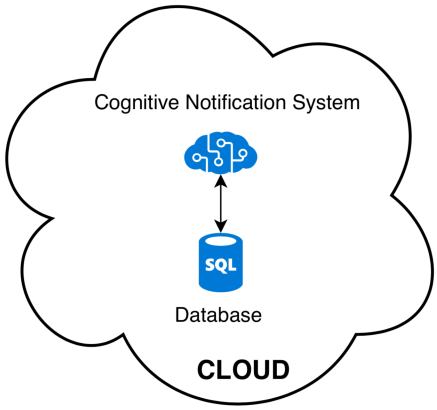


Figure 3.3: System's cloud with focus in the Cognitive Notification System and database interactions

3.1.2 Cloud

The cloud role in this work is to be the center of operations of all the system, as shown in figure 3.4. It has the capability of managing all the smartphone application and smart lock hardware HTTP requests, store all the activity generated and users data and, at the same time, it runs the algorithm that triggers the application cognitive notifications.

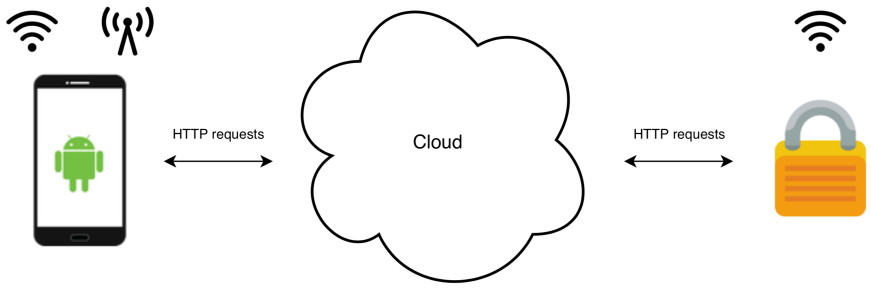


Figure 3.4: Cloud communications diagram

3.1.3 Hardware

The physical smart lock prototype is composed by:

- Microcontroller with wireless communications module;
- Door lock actuator;
- Door position sensor;
- Presence sensor;
- Light Emitting Diode (LED).

The microcontroller is programmed to connect, through Wi-Fi technology, to the web in order to receive users unlock requests, to perform lock and unlock actions through the door lock actuator and, at the same time, read the door position sensor to understand if the door is opened or closed and the presence sensor so that the system knows if there is anyone at the door where the smart lock is assembled, this helps the cloud to register the user's activity and adds a security layer by preventing unlock actions when there no one at the door. The smart lock hardware prototype connects directly to the cloud through HTTP requests as figure 3.5 shows and operates as shown in figure 3.20.

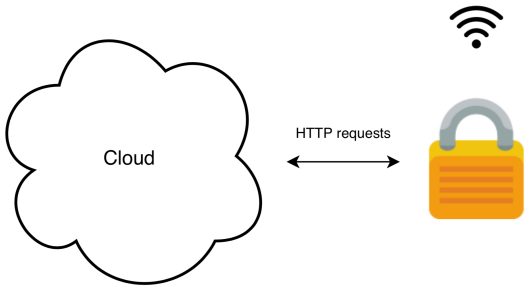


Figure 3.5: Smart lock hardware communications diagram

3.1.4 User/Smartphone Application

Through the smartphone application, the user can perform several actions. After signing up and while logged in, the user should be able to control the door lock mechanism in order for it to perform an unlock action. It's also possible to check an activity log where the ten last smart lock activities are listed and to change personal data such as the username, first name, last name and the password. Summing up, the smartphone application has the following features:

- User register/login/logout;
- User profile edit;
- Smart lock unlock action control;
- Smart lock activity log.

This smartphone application has been developed for Android-based smartphones and its operations are based in Hyper Text Transfer Protocol (HTTP) requests that are made to the cloud through the web as shown in figure 3.6.

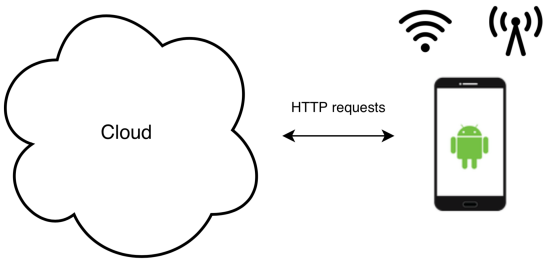


Figure 3.6: Smartphone application communication with the cloud diagram

3.2 System model

Section 3.1 is focused on describing the functionalities of the work's individual modules to gain a greater perception of each one's role in the whole system.

This section aims to describe the whole system behaviour in a more detailed way, in order to properly modulate and implement each one of this work's modules and to pursue an easy integration between them. Describing the system's behaviour is also important to understand the interactions between all the modules, and the flow of actions that define the correct system implementation.

As explained in section 3.1 the user controls the smart lock hardware through the smartphone application with the cloud acting like the middleman that coordinates all the system requests between all the modules. That means that everything that happens in the whole system, represented in figure 3.7, is controlled, checked, registered or even authorized by the cloud.

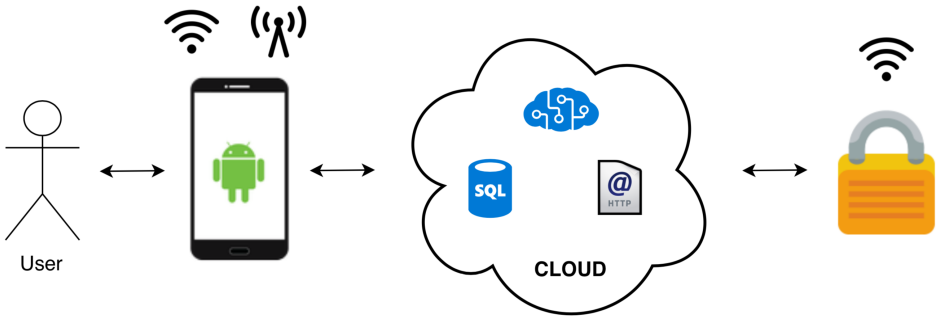


Figure 3.7: Cognitive Notification System

To better describe the different scenarios where the modules that take part in the system interact, five sequence diagrams were used. Sequence diagrams are one of several different Unified Modeling Language diagrams and, are used with the purpose to describe and show the interactions between objects in a sequential order that those interactions occur, [32]. Sequence diagrams are designed to provide a general flow of what can happen in a system during a regular system usage, without focusing on the details and specifics of what's happening in each module. The diagrams in figures 3.8, 3.9, 3.10, 3.11 and 3.12, were designed with the fact that all the systems are up and running in mind.

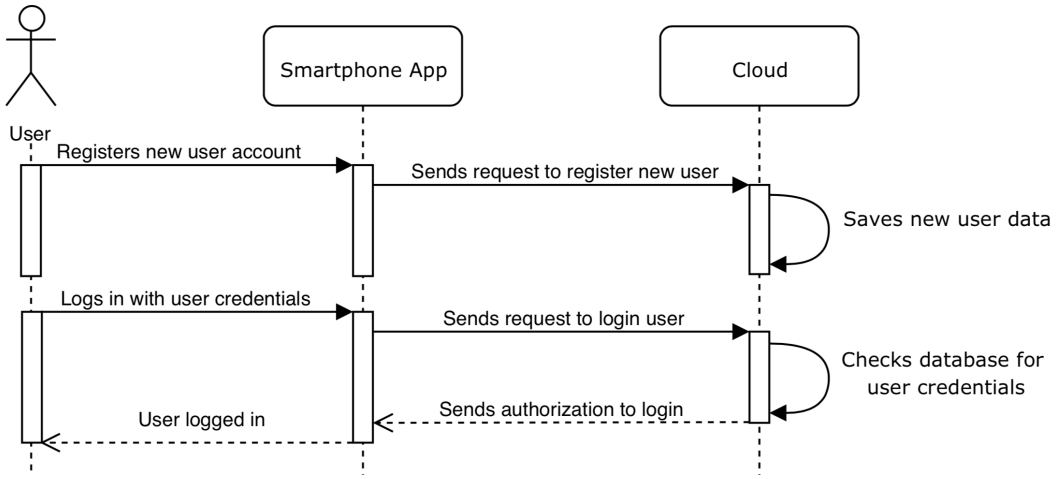


Figure 3.8: Register and login sequence diagram

The following diagrams assume that the user is already registered and only needs to login in order to perform different actions.

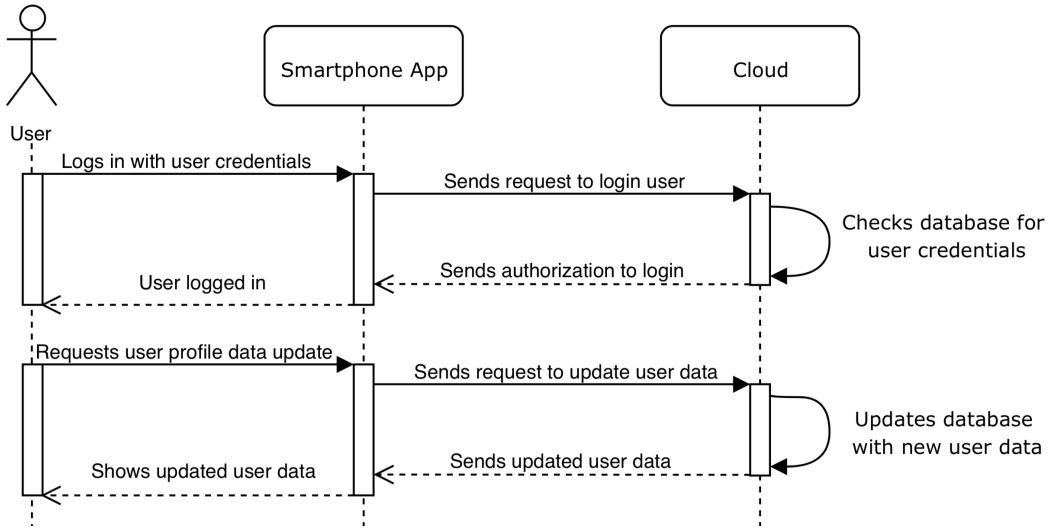


Figure 3.9: Check profile data sequence diagram

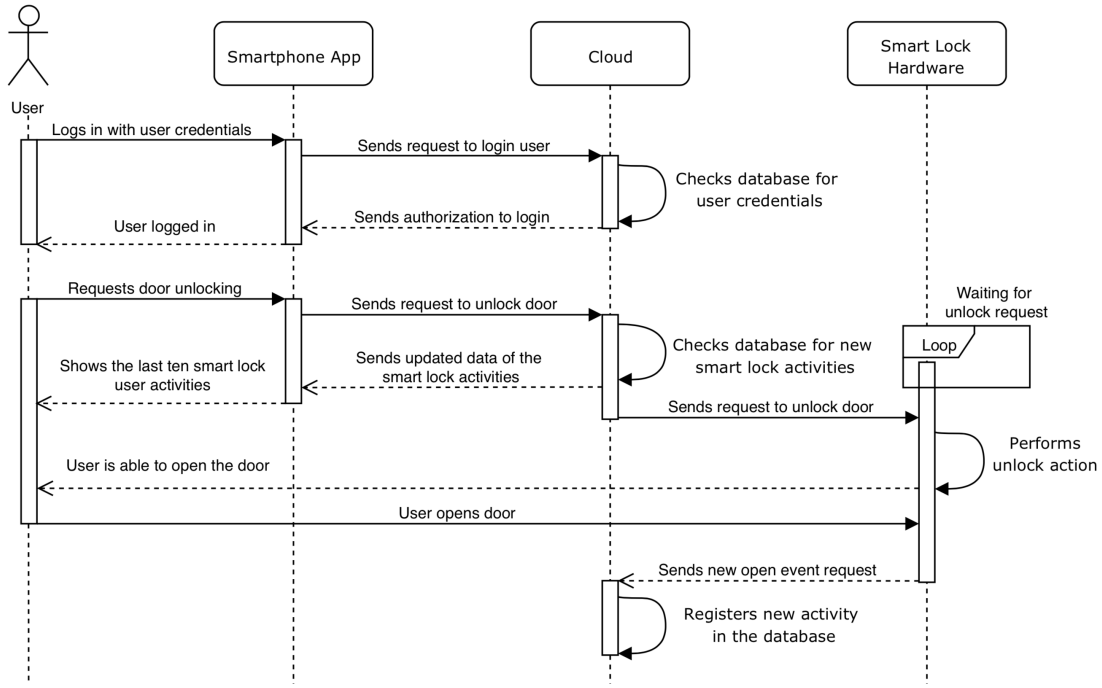


Figure 3.10: Unlock action sequence diagram

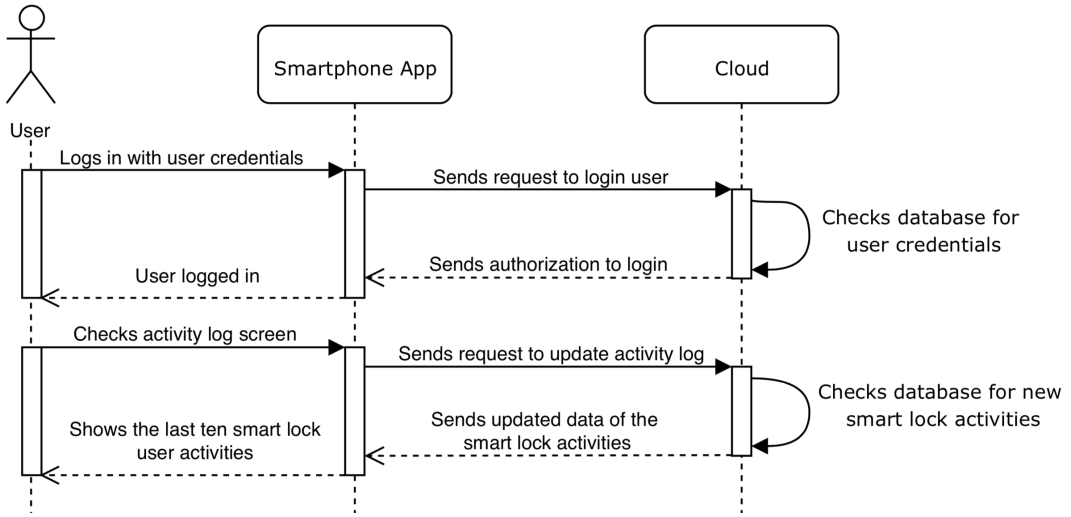


Figure 3.11: Check user activity sequence diagram

The sequence diagram shown in 3.12 was designed considering the fact the the system administrator doesn't need to be logged in to receive the cognitive notifications in his smartphone.

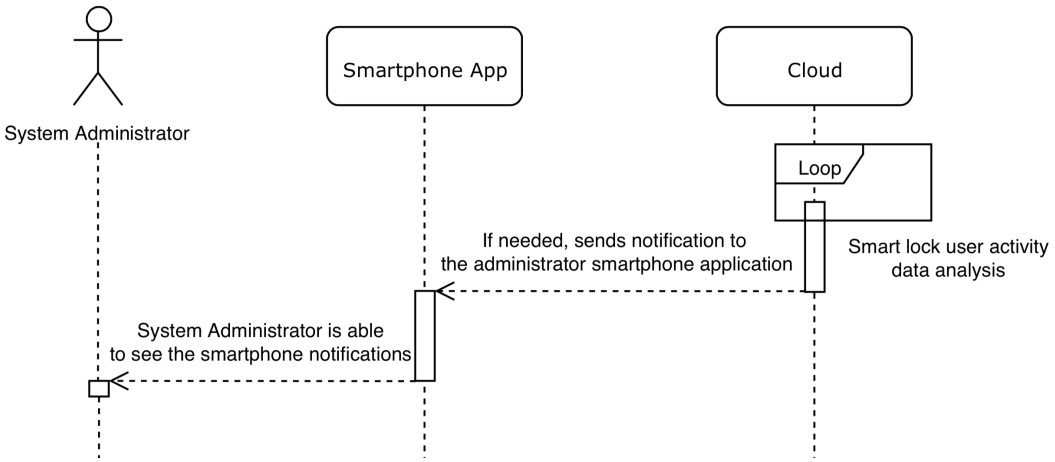


Figure 3.12: Cognitive notification system sequence diagram

The next section approaches the detailed operations of each system module and functionalities.

3.3 System functions

Now that the whole system model is described in section 3.2, with the help of the sequence diagrams in figures 3.8, 3.9, 3.10, 3.11 and 3.12, it's important to define and describe the specific behaviour for the system's different functionalities briefly mentioned in 3.1.

3.3.1 Cognitive notifications function

The cognitive notifications function runs entirely in the cloud platform. It analyses the data generated through the user's interaction with the smartphone application and consequently with the smart lock hardware and creates enough knowledge to understand each user routines and fire the notifications to the system administrator smartphone application. It all begins with the user performing several unlock actions through the smartphone application in order to unlock the door and enter the building where the smart lock prototype is assembled. By gathering the smart lock usage data in the cloud platform database, the notification algorithm is able to predict when a given user is going to unlock the door and by having a next unlock action predicted time the system will wait for that given user to unlock the door within a time window. If it happens, the new activity is added to the database and used to recalculate the new predicted unlock activity, if not, the notification, warning the system administrator that the user is late, is fired to the system administrator. The notification is only sent after a trigger time, thirty minutes, that is defined in the cloud cognitive notification algorithm. That trigger

time defines a gap between the user predicted time of arrival and the notification being sent to the system administrator. This specific mechanism prevents the arrival of a great number of unnecessary notifications to the system administrator smartphone. Another important feature that allows the system to perform in a better way, is the fact that each day is divided in three periods. This prevents that an unexpected unlock and entry action distorts the smart lock usage and eventually mess up the notifications send timings. The cognitive notification function is described in figure 3.13 and was designed assuming that the system's database has enough user activity data to process.

Cognitive notification trigger time arrived?

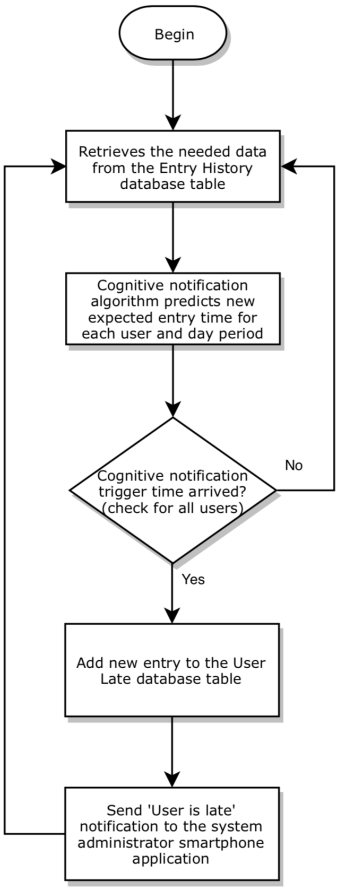


Figure 3.13: Cognitive notifications function flowchart

3.3.2 Register function

The smart lock user-to-be has to register himself in the system in order to use it. The register action is performed, by the user-to-be, through the smartphone application by inserting his own personal data and pressing the button that proceeds with the action that creates a new user account. After this action, the inserted data is verified in the cloud and if everything is right the new user is now free to login in his personal smartphone

application account and able to freely use the smart lock application functions. The register function is described in detail in the following figure 3.14 and was designed assuming that the user-to-be is already in the smartphone application register screen.

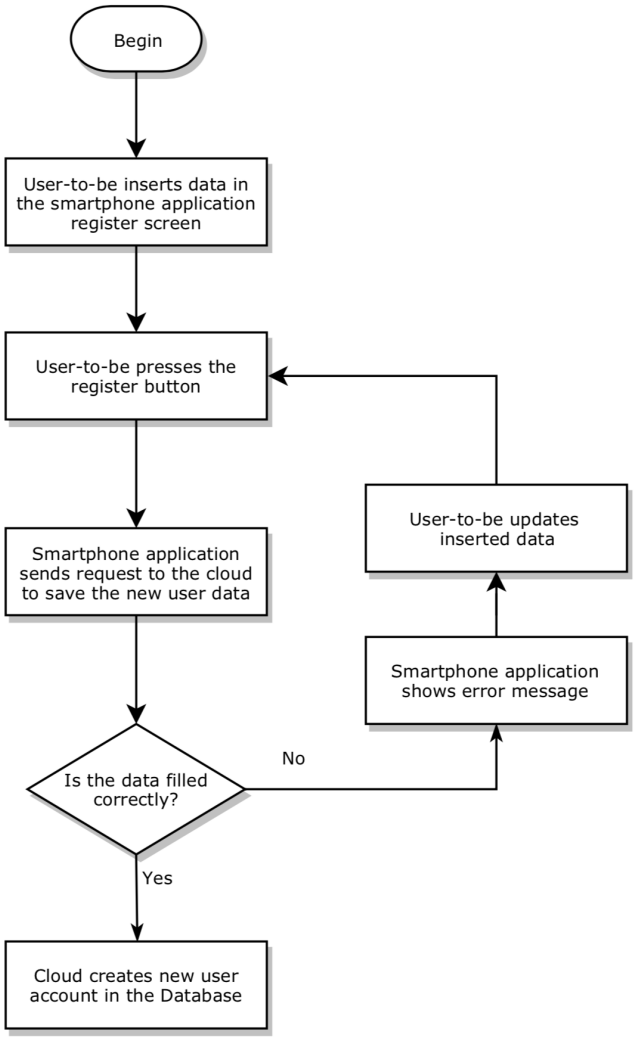


Figure 3.14: Register function flowchart

3.3.3 Login function

After registering himself, the user is able to log in and access all the smartphone application functionalities. In order to login into the app, the user has to fill the correct credentials for the application to know who is logging in and press the login button. After pressing the login button the cloud receives the login request and verifies if the credentials are correct. If the answer is positive, a new user session is created. If the user inserts credentials that don't match the credentials of a registered user, it receives a login error message. The login function is described in the following figure 3.15 and was designed assuming that the user that is about to log in is already in the smartphone application login screen.

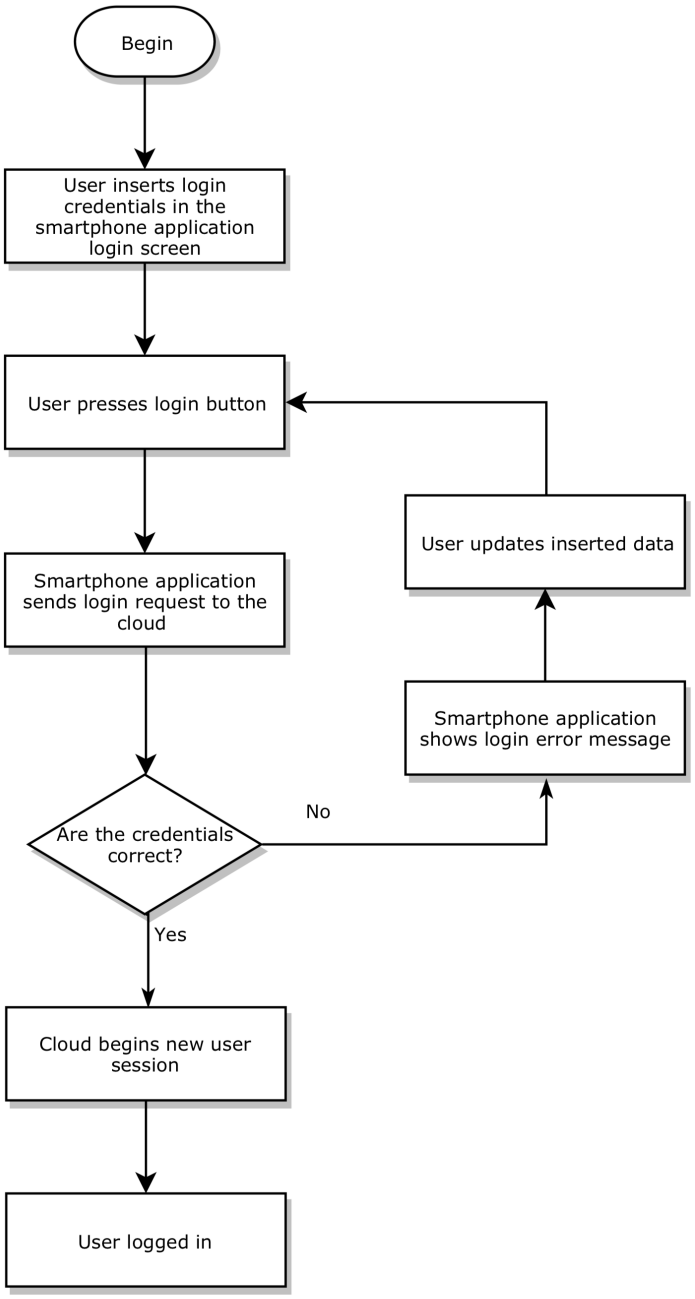


Figure 3.15: Login function flowchart

3.3.4 Logout function

Logout function is the simplest one of this work. The logged in user has the possibility to log out from the smartphone application from several differente screens. This action can prevent unknown people to access our smart lock smartphone application and consequently to gain control over the smart lock itself. The logged in user that wants to log out from the smart lock smartphone application has to press the logout button. After that, the smartphone application leads the user to the login screen preventing anyone to control the system through the application. This flow is shown in figure 3.16 and was designed assuming that user is already logged in.

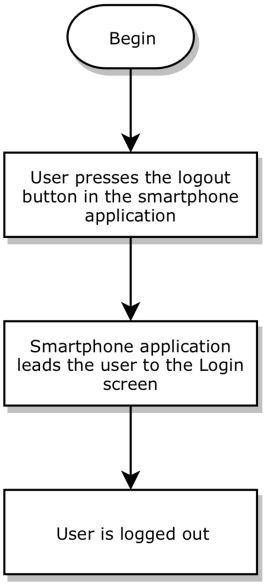


Figure 3.16: Logout function flowchart

3.3.5 Edit user profile data function

Smart lock users have also the possibility to change their own personal data through the user info screen as focused in 3.1.4. For the user to accomplish a successful change of their personal data in the system, they must go to user profile screen and change the field, or fields, that demand a change and press the save changes button. This action will trigger an update user data request into the cloud, that will save the new user profile data in the database. This process is described in figure 3.17 assuming that the user that wants to change his own personal data is already logged in and in the user profile screen.

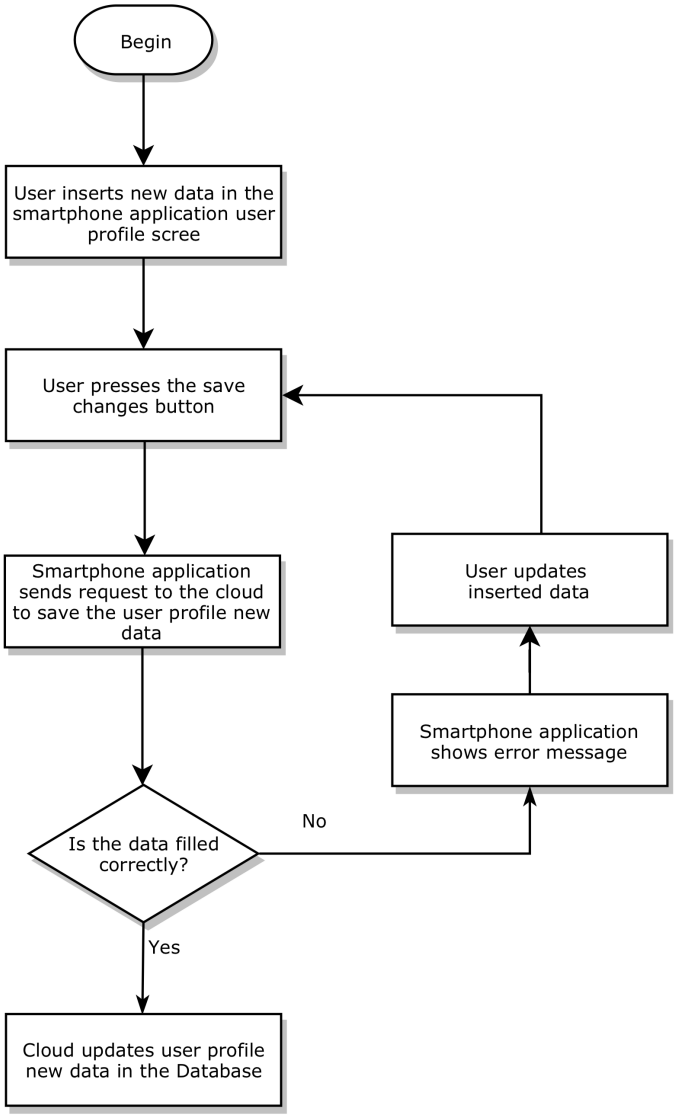


Figure 3.17: Edit user profile data function flowchart

3.3.6 Activity log function

The system smartphone application gives the users the possibility to check their own smart lock system activity log. This function aims to show the user the last ten smart lock unlock actions that were performed by himself. It features the user's username and a timestamp (date and time) that shows when it was performed. As mentioned, the users can only see their own activity, although the system administrator can check not only his own activities but also the other users activities, this is limited to the ten last unlock actions. The process that leads to displaying the data in the smartphone application is described in figure 3.18 that was designed assuming that the user is already logged in.

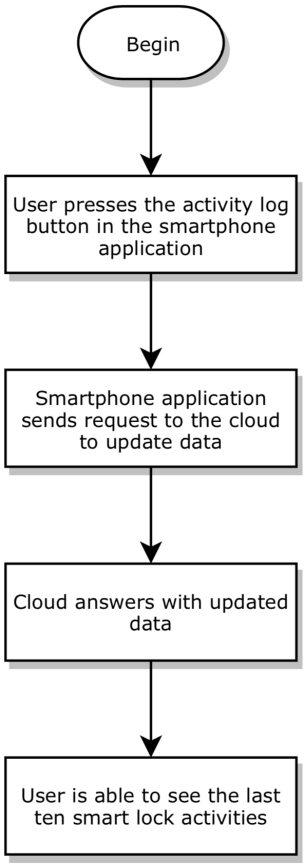


Figure 3.18: Activity log function flowchart

3.3.7 Unlock function

Unlock is one of the system's core functions. This functionality makes it possible to gather the smart lock activity data that is used to generate the cognitive notifications that are fired to the system administrator. To use this specific feature, the user must be at the smart lock screen in the smartphone application and press the unlock button in order to send a request to the cloud that will give the hardware authorization to unlock. After this initial process, the magnetic sensor will read the door state aiming to understand if the user opened the door, or not, during the next thirty seconds. If the user opens the door, the hardware sends a request to the cloud to save a new log into the smart lock activity database, if not, the smart lock locks the door automatically. The detailed process of unlocking the smart lock is described in figure 3.19 and was designed assuming that the user is logged in and is already in the smart lock screen in system's smartphone application

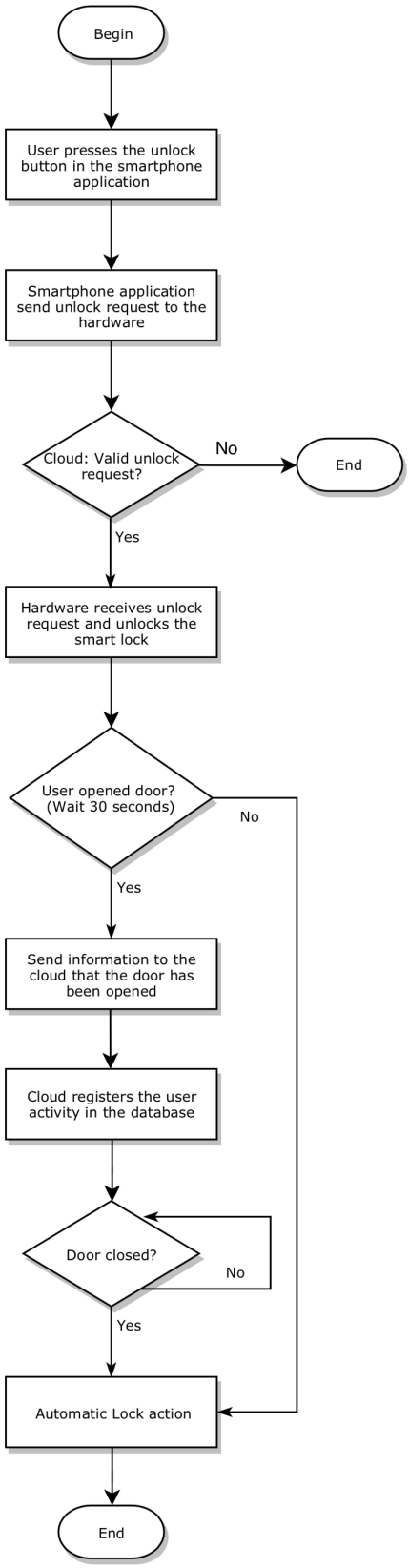


Figure 3.19: Unlock function flowchart

3.3.8 Smart lock hardware

The hardware module plays a major role in the whole system, despite the fact of not being a function on its own, it uses different functions to succeed in the tasks that are assigned to it. The hardware represents the smart lock itself, it helps the system to have a physical representation of the unlock action itself and is extremely important assuring that the generated data is quite reliable. As mentioned in 3.1.3, the hardware is composed by the a microcontroller with wireless communications, a door lock actuator, a door position sensor and a presence sensor. To work properly, the hardware starts to execute a calibration routine to assure that the door lock is in the lock position and connects itself to the predefined Wi-Fi network. After this, it stands still until an unlock request from the cloud arrives. After the request arrives the presence sensor turns on so the system if there is anyone around, if yes the door unlocks and the green LED turns on to let the user know that the door is now unlocked. After, the door position sensor waits for the door to open in order to send the cloud the permission to register a new user activity in the database. If the door is opened the activity is registered. If not, the door locks automatically after thirty seconds. The door position sensor also helps in avoiding unnecessary lock/unlock actions. The detailed operations of the hardware module are described in figure 3.20.

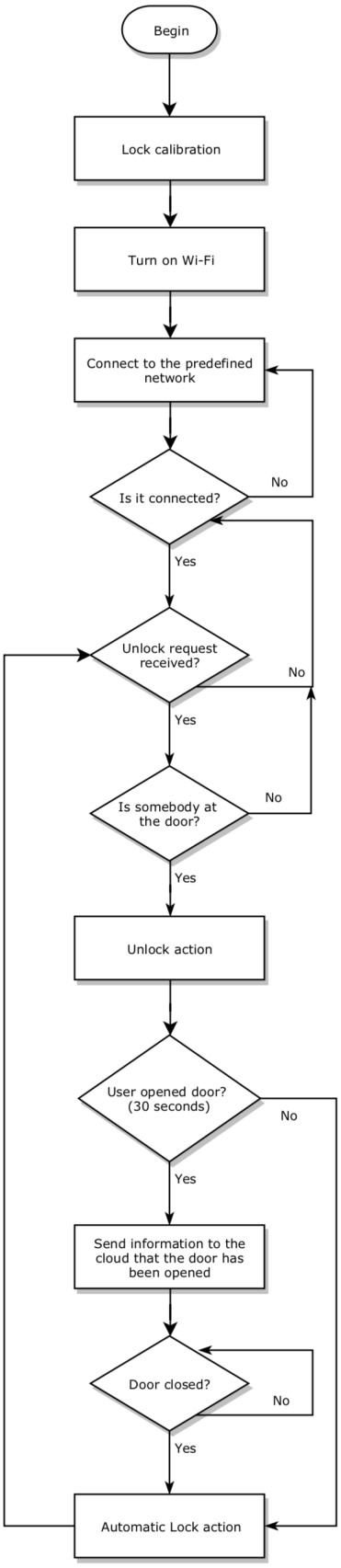


Figure 3.20: Smart lock hardware flowchart

3.4 Data model

To support the, previously approached, system functions and operations, this work needs to have a solid and strong system data model. This data model is able to store different categories of data that have different purposes and roles throughout all the system's operations. In this section are itemized and described the different data sets that are needed for the system to work as previously described in sections 3.2 and 3.3.

- **User** - User data table stores the personal data of all the system users. Each user is identified through an unrepeatable Identifier (ID) and the personal data is also composed by the user's first name, last name, username and password.
- **Role** - Role table stores the two different system roles, regular user or system administrator. This table is connected directly to the User table in order to identify each user's role.
- **Lock** - Lock database table ensures that all the requests that travel through the system are connected to the correct smart lock. This is guaranteed by generating a unique ID, that identifies the physical hardware, of each one of the smart locks that may exist in the network. This avoids users to communicate with other smart locks, that they shouldn't have access to since the requests are directed only to a specific one. Despite the fact that the developed system has only one lock, it's important to have this table in order for the system to grow in future developments. This table is composed by a lock ID that identifies the smart lock hardware and a variable that stores the name of the lock itself. It connects to the Unlock Request and Open History tables.
- **Unlock Request** - This table receives an entry every time an unlock request is made and since the unlock request is valid for only thirty seconds it's mainly used to check if the unlock request has expired or not. The table is composed by a request ID, a datetime variable that stores the date and time in which the request was made, and connects to the User and Lock tables in order to link request with a specific user and lock.
- **Open History** - The Open History table has the main task to save all the user's activity history. To do that, this table stores the date and time of each user's unlock and entry action and assigns each one of them to one of the three available day periods. This helps the cognitive notification system to segment the notifications. At the same time, it connects to the User and Lock table to assign a specific user and lock to that unlock action.
- **Entry History** - This table is very similar to Open History table, the main difference is that Entry History table only allows each user to have registered an unlock and entry action per period per day, so that the system administrator smartphone doesn't

get flooded with unnecessary notifications. This table stores the day of the year (from 0 to 365) and the period of the day in which the unlock and entry action was registered. It also connects to the User table in order to link the registered action to a specific user.

- **User Late** - The cognitive notifications algorithm studies the data in the Entry History table every minute, if the user is late, an entry in the User Late table is created. The creation of a new entry in this table trigger the notification send to the system administrator smartphone. This table connects to the User table where it obtains access to the user ID. Since that table is also connected to the Entry History table, it is possible to access to the day of the year value and to the period of the day in order to maintain full register of the sent notifications.

Chapter 4

System Implementation

This chapter describes the system development and implementation of the whole system different modules.

Section 4.1 introduces the cloud application development, presents the development of the Cognitive Notification System and every functionality that runs on the cloud platform.

Section 4.2 is focused on the smart lock hardware assembly and programming.

Section 4.3 describes the smartphone application development.

4.1 Cloud

The application that runs within the cloud server in the AWS (Amazon Web Services) is the core of the system. It is developed through Java programming language and it uses the Spring Boot framework that enables a faster application development and deployment.

Spring Boot is a facilitator that helps the developer to focus on the developing process itself, leaving configurations and other secondary tasks behind. Through simple tags, or annotations, Spring Boot lets the developer define specific behaviors and functionalities for each Java class that is created and consequently for the methods defined within those classes.

Using Spring Boot framework makes it easier to create the application database, the tables within and even the relationships between, only by adding specific framework tags related to the database. Also, there are some already developed classes and methods which can be used to develop the application faster.

The same can be applied to other branches of the application. Despite the fact that this work uses OAuth 2.0 authentication system, Spring Boot framework includes it's own

system that is easily deployable, with ready to use classes and methods that can suffer further development in order to meet the application goals.

Despite the fact that most of them had suffered further developments, using Spring Boot also enabled the usage of several already built classes and methods that have helped the application to meet it's goals in a faster way.

After creating an account at the AWS platform, a server running Linux Operating System was chosen. Java software was installed and the last step was putting the *.jar* file, containing the developed application, running.

4.1.1 Cognitive notifications system

The Cognitive Notifications System is developed within the cloud platform and gets the most of the other cloud developed modules so it works in the most reliable way.

This section describes the Cognitive Notification System development in order to fully understand what is happening behind the scenes. It is divided in three different parts, cloud database, system requests and communication, and security.

The developed system is built in order to generate data that can feed the algorithm that calculates the user's routines.

The cognitive notifications system is based on previously acquired knowledge, in this specific case is based on the previous users activities with the goal to establish routines and predict the next users unlock and open actions.

To study the data there is a need to store it first, so every smart lock unlock and open activity is saved in the Open History table. Every time an entry is created in this table it is identified through a value that assigns a defined day period for that specific activity as shown in table 4.1.

Table 4.1: Value assigned for each defined day period

Day period	Value
08h00 - 14h00	0
14h00 - 20h00	1
20h00 - 08h00	2

These values are assigned to an integer variable, named "period" , in the Open History database table in order to segment each user activities. This segmentation intends to remove any distortions from the user's different routines so that a morning (between 08h00 and 14h00) unlock and open action does not have an influence in the user night (between 20h00 and 08h00) activities.

When a specific user performs an unlock and open action a new entry is created in the Open History table and if it is his first activity in that day and period another entry is automatically created in the Entry History database table. This means that each user only has a maximum number of three entries in the Entry History table per day, one for

each day period. This happens so that the system can select the user main activity, in each period, from all the others. For example, a specific user arrives at his own home at 18h37 and repeats the action at 18h55 because he left his home to walk his dog, despite the fact that both situations are considered as unlocking and open activities, they should not have the same meaning and weight for the system calculations, that's why the Entry History table only saves the first activity per period per day of each user.

Entry History table is directly used to understanding each user's routines and to deploy the notifications to the system administrator, because the data within this table is used to the calculations that provide an expected time for each user next activity.

The next activity expected time is calculated every minute, using the Entry History table, for every user that has at least five unlock and open activities in the database and at least one of them in the last five days. The fact that the system only uses the last five activities gives the system flexibility in order to adapt to the user's routine changes. It also helps the system administrator to not receive unnecessary notifications referring users that do not use the system that often. The expected next activity time is calculated through the average of the last five activities. Relating this calculation with the periods of the day in which unlock and open actions happened in the past, the system has the ability to define a delay, that is related to the hour in which each period begins. Every minute the system is looking for new activities to make new expected action calculations, but if that new activity does not happen and if, through the calculations, a certain user reaches the delay time (measured in seconds) plus the thirty minute notification trigger time, the system creates a new entry in the User Late table.

When a User Late table entry is created, the system knows that is time to send a notification to the system administrator smartphone. Google Firebase [33], a very broad service with a great focus in analytics and cloud messaging, is used to send the notifications to the system administrator smartphone application. When the notification trigger time is reached and a new entry is created at the User Late table, the system connects to Firebase platform and, a minute after, the notification is sent indicating that a specific user is late. The system administrator smartphone application is the only one who receives the notifications since it is the only one who is subscribing the Firebase topic through which the notifications are sent. Examples of the notifications received by the system administrator are shown in figure 4.1.

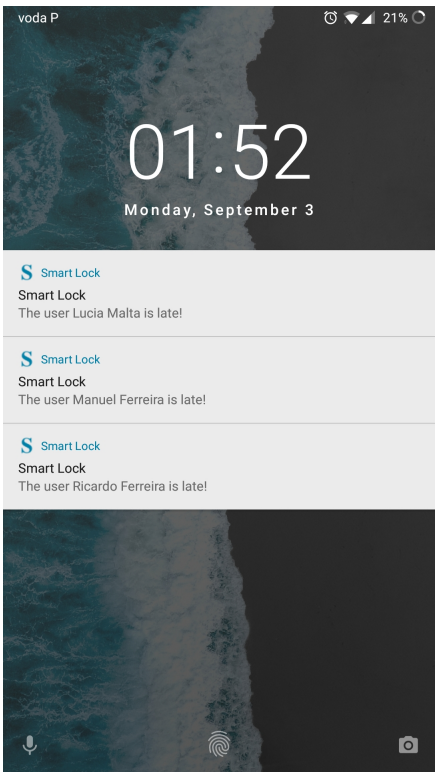


Figure 4.1: Cognitive notifications in the system administrator smartphone

4.1.2 Cloud Database

The cloud platform presented solutions in 2.2.2 include a free trial period with limited GET and PUT movements, with AWS and IBM Cloud offering the biggest number of those movements. Considering the research done on this technologies and also the quality/quantity of online support for beginners in each one of the presented solutions, the author selected Amazon Web Services cloud platform to develop this work.

The algorithms running in the dedicated server at Amazon Web Services cloud platform are responsible for maintaining all the system up and running and, at the same time, a stable connection that allows the data to flow through the system.

The system's cloud database it's one major part of this work, since it support a lot of all the system's functions including the cognitive notifications system.

The system's database is built in Structured Query Language (SQL) widely used to store and manipulate data in databases, and it is hosted in the dedicated server used for this work at the AWS cloud plataform. The database tables and relational model were built with the system specifications in mind. The database relational model developed for this work is as shown in figure 4.2.

User table consists in saving the users personal data with a focus on the User ID and Username variables that identify the users. Each one of those variables is defined as Primary Key (PK) so that the value that they save is unique. Role ID is a Foreign Key (FK) in this table so that the system is able to identify the user as a regular user or a

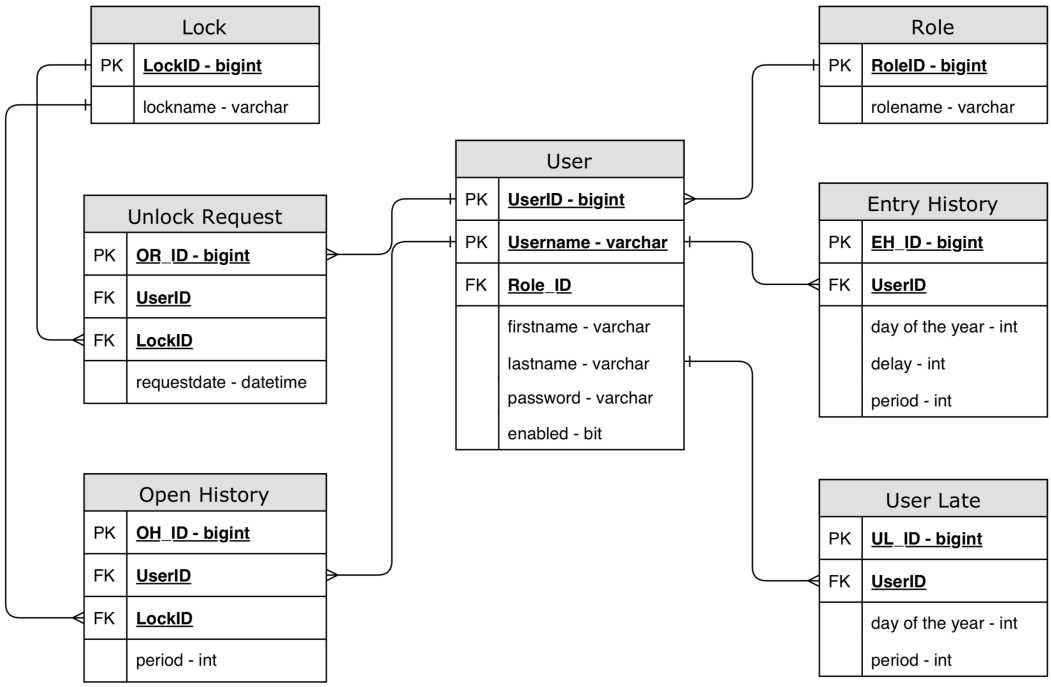


Figure 4.2: Database relational model

system administrator. To know if a specific user has already been activated by the system administrator, a variable named Enabled is used. It has two possible values, zero for a user that it is not active yet, one for an active user. User table is connected to almost every other database table. It has a many-to-one relationship with the Lock table because many users can have the same role but each user only has one role. At the same time, it has a one-to-many relationship with other four tables, Unlock Request, Open History, Entry History and User Late linking the users to all the activities that have been performed and consequently saved in the database.

The Unlock Request is the table that saves all the unlock requests made by the users through the smartphone application. Is linked to the User table in order to identify who made the request, and since a user can make as much unlock requests as he wants, this table has a many-to-one relationship with the user table. This table is also connected to the Lock table, through a many-to-one relationship in order to identify, through the Lock ID FK, the smart lock hardware for which the request was made. This table also features a DateTime variable, named request date, to check if the unlock request is still valid or not. This avoids that the smart lock hardware performs unlock action from old unlock requests.

The Open History table is similar to the previous one but, this one only saves the unlock actions when they are followed by an open action. This table features the same FK as the previous one in order to identify the user who performed the unlock and open action and also to identify the smart lock hardware that was used. It also has a variable named Period in order to save the period of the day in which the action was performed,

as shown in 4.1.

The Entry History table has its entries saved at the same time that the Open History table with the difference that this table only stores a maximum of three daily unlock and open action for each user, one per each period of the day by using the Period and Day Of The Year variables. This table is the one that is used in the cognitive notifications calculations because it saves the relative delay, in the Delay variable, that a specific unlock and open action has to the beginning of the period in which that action happened.

User Late table is the one responsible for storing the data needed so that the cognitive notifications are sent to the system administrator smartphone. It features User ID as a FK to identity the user and fetch his data, needed to deploy the notification. Is also features the Period and Day Of The Year variables so the system doesn't duplicate notifications to send.

4.1.3 System requests and communication

The system requests are essential to the system's modules communications and interactions. System requests support functions as when a user logs in or unlocks the door through the app, or even when the smart lock hardware notifies the cloud platform that the door has been opened in order to register a new user unlock and open action in the cloud database. HTTP requests are used to establish communication between a client and a server and have several different methods, this system requests are based on GET and POST methods. The GET method as the function to request data while the POST method has the function of sending data, [34]. There are HTTP requests involved in almost every system action, those requests are described in this section.

4.1.3.1 User related requests

- Create User - This POST request is triggered when a new user registration happens in the smartphone application. It sends the data, that was inserted in the application Register Screen, to the cloud. A username existence check is made in the user's database in order to avoid having two users with the same username. If the username does not exist a new entry in User database table is created with the data that has been sent to the cloud through this request. This request does not require a session token.

- Activate User - As the request name indicates, this request has the purpose to activate a newly registered user. Despite the fact that this request is not implemented into the system administrator smartphone application as it should its used to activate registered users accounts before they can perform a login action that would give them access to all the smart lock system functions. This adds up a security layer to prevent a stranger to gain access to the system. During the system implementation and testing, this request was triggered through the Postman software.

Since only the system administrator can use this request the session token used in Postman to send this request had to be the administrator session token.

- Login - The login request happens when the user inserts his own credentials in the Login screen at the smartphone application and presses the Sign In button. After this, the credentials are sent, with the POST method, to the cloud through an OAuth Token Request [35] that answers, if the credential matches the database, with a session token that will allow the user to proceed into the using the smartphone application features.

- Get User - This GET request is used in the smartphone application to show the logged in user data in the Profile screen where the user is able to see their own personal data. This request answer includes the user's own username, first name, and last name. For this request to show the user's own personal details instead of another user details, it has to be identified with the session token that is created when the user logs in.

- Update User - This request updates the user data in the cloud database. It is a POST request that sends the updated user information (first name, last name, and password) from the smartphone application Profile screen to the cloud where is stored in the User database table. The user who sends the update request is identified with the session token that is created when the user logs in so that the updated user data is stored properly.

- List History - This GET request is used to retrieve all the smart lock unlock and open activities history. This request is used in the smartphone application Activity Log screen that then only shows the last ten user activities. This request retrieves the smart lock open date and time, the user who performed the action full name and username, the lock name and Identifier (ID) and a timestamp. This request requires a session token in order to prevent that unknown to the system people can access to the system data.

- List Users - This request uses the GET method and retrieves all the registered users. The requested answer lists all the user's usernames, first names, and last names. Despite the fact that this request is not used in any of the system functions it was used during the system testings in Postman software. This request also requires a session token so that unknown persons to the system don't get access to all the user's data.

4.1.3.2 Smart lock related requests

- Unlock Event - This is the GET request that is sent through the smartphone application every time a user wants to unlock the door in which the smart lock hardware is assembled. This request returns a true value every time it is triggered. A session

token is needed in order to perform this request, this avoids unknown to the system people to get access. Also, this request's Uniform Resource Locator (URL) needs to have the Lock ID included so that the request is made to the right lock, despite the fact that in this case that would not be a problem since the system only has one lock connected.

- Lock State - This GET request is used in the smart lock hardware routine to know if the Unlock Event request has been made or not. It returns a boolean value, true if an unlock request has been made and is still valid, false if there is no valid unlock request pending. The smart lock hardware unlocks itself when it receives the true answer to this request. This request does not require a session token since it only indicates if there is a valid unlock request or not, but it does need the lock ID in the request URL to identify the lock for which the request was made.

- Open Event - When the smart lock hardware unlocks and it senses, through the magnetic sensor, that door has been opened, it sends this request to the cloud indicating that someone opened the door by sending the true boolean value. After this, the cloud registers a new entry in the Open History database table with the data from the user who made the last Unlock Event request. This request does not require a session token since it only gives the information that the door has been opened, but it does need the lock ID in the request URL to identify the lock for which the request was made.

- List Locks - This request uses the GET method and retrieves all the registered smart locks. The requested answer lists all the locks information ID, name and the lock state, true for unlocked, false for locked. Despite the fact that this request is not used in any of the system functions it was used during the system testings in Postman software. This request also requires a session token so that unknown persons to the system don't get access to all the system's locks data.

4.1.4 Security

As already stated, this work's communications happen in the web through HTTP requests. That HTTP requests transport data with them making them dangerous to fall into unknown hands.

In order to add a security layer to the system communications, this work uses OAuth 2.0 as an authorization protocol that helps the system to know who has the authorization to make the requests needed for this work communications. OAuth 2.0 is the actual industry-standard protocol for authorization and communication with client and server [36], it helps this work's communications to maintain a certain security level that is indispensable in the Internet of Things (IoT) context.

When a user logs in into the smartphone application, it sends the credentials to the cloud platform that validates, or not, those credentials. If the user credentials are

correct, the cloud platform acknowledges that a user logged in and, since the OAuth 2.0 configuration is in the cloud, it assigns an Access Token to that user, as shown in figure 4.3, that will be present in every request that the logged in user will do afterwards. This Access Token, named as session token in this document, as a duration of eight hours in which the user can make as many requests as he wants. After those eight hours the user has to perform another login action in the smartphone application. This process makes it unnecessary to exchange the user credentials everytime a request is made and also helps to identify the user that is making the requests because each key is unique.

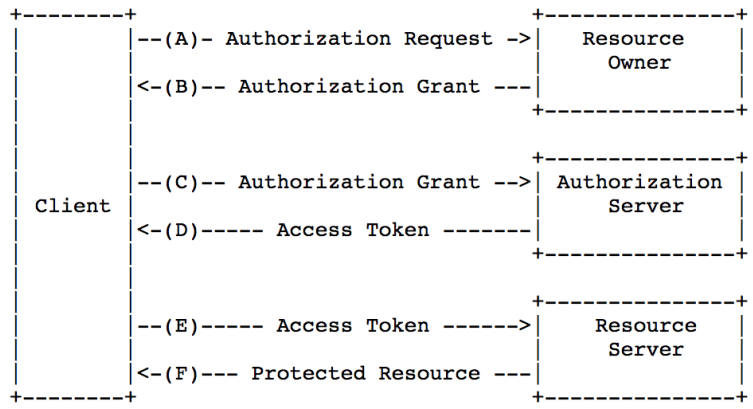


Figure 4.3: OAuth 2.0 protocol flow taken from [37]

4.2 Smart lock hardware

To develop and implement the physical smart lock prototype there is a need to correctly assemble all the necessary sensors and actuators to the chosen microcontroller. The system is composed by the following components:

- ESP8266 development board;
- Pyroeletric infrared (PIR) sensor;
- Reed switch magnetic door sensor;
- Direct Current (DC) servomotor;
- Green Light Emitting Diode (LED).

The ESP8266 development board was chosen because it features a reliable Wi-Fi microcontroller, it's cheap, its programmable with Arduino Integrated Development Environment (IDE) in which the author had previous experience and has enough GPIO pins for this work needs. The ESP8266 development board controls all the hardware operations. It manages the reed switch and PIR sensor readings, it controls the DC servomotor movements and, at the same time, it communicates, through the web using

the ESP8266Wifi.h library [38], with the cloud platform in order to manage the unlock and entry register requests. The ESP8266 has a great advantage of being programmable through Arduino IDE which helped in the hardware development since the author had previous developing experience with this software, [28] [29].

PIR sensors help security systems in detecting humans presence through infrared radiation also they are cheap, easy to use and has an easy data processing. The PIR sensor enables the feature of the smart lock prototype to detect, or not, movements from heat sources, with a great focus on humans. This creates an additional security layer since the door can only be unlocked through the smartphone application if this sensor detects presence at the door in which the smart lock is assembled. This sensor is defined as an input and since it has an operating voltage between 4.5 V and 20 V it is connected to the ESP8266 Vin pin in order to use the Universal Serial Bus (USB) port voltage of 5 V. It also has to be connected to a ground (GND) pin and finally to a digital pin so that ESP8266 is able to digitally read the sensor value. The sensor's output varies between LOW (0) no presence detected or HIGH (1), as shown in figure 4.4 that means that a presence has been detected, [39] [40].

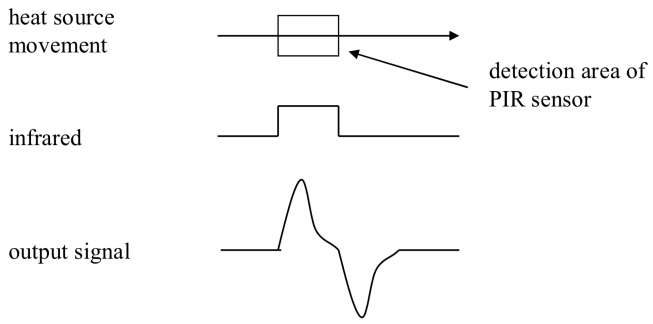


Figure 4.4: PIR sensor signal output taken from [40]

The smart lock prototype uses a sensor in order to read the door's position, opened or closed. Reed switch magnetic sensor was chosen to perform that action. Data processing with reed switch sensor is easy since it has only two possible outcomes that indicate if a door is opened or closes. Also, this type of sensors is widely spread for window and door usage monitoring.

Knowing if the door is opened or closed it's important to avoid unnecessary unlock-/lock actions but also for register only the true door activity. The sensor that is used in this work's smart lock prototype is in Normally Open (NO) mode. This means that when the switch is not affected by a magnetic field, the switch is opened and then, the current isn't able to flow between the two terminals. The opposite happens when a magnetic field is near the switch, the contacts close and current flows, [41]. This sensor is connected to the ESP8266 through a digital pin and to a GND pin meaning that when there is a magnetic field nearby the switch closes connecting the digital pin do the GND making the microcontroller read a LOW (0) value. When the magnetic influence is moved away the

switch opens and the microcontroller reads floating values. To avoid those floating values the digital pin where this sensor is read is defined as an input pull-up that ensures, by using an ESP8266 internal resistor, that when the reed switch is not closed the input pin reads a small amount of current that is flowing between the Voltage Common Collector (V_{CC}) making the microcontroller to read an HIGH (1) value, [42]. This makes it easier for the system to understand when the door is open or closed.

Since the smart lock hardware prototype will only be used as a proof of concept of the overall system, the selected servomotor doesn't need to obey to specific requirements. TowerPro SG-90 servomotor has an operating voltage around 5 V but it can be used with and a rotation amplitude of 180, it only weights 9 g and it's an easy hardware to find and buy since it's commonly used in small prototyping projects. The servomotor has three wires to connect to the ESP8266, the power supply wire that it's connected to the 3.3 V pin, a GND wire and the control signal wire that is used to control the servomotor position. Despite the fact that the servomotor datasheet [43] refers 5 V as the typical operating voltage, the servomotor can be powered at 3.3 V but it eventually performs with less torque than the stated at the datasheet. To control this actuator, Servo.h library was used [44]. During the regular functioning of the hardware, the servomotor performs a calibration routine before the microcontroller connects itself to the defined network, this routine consists in a unlock and lock action in order to assure that smart lock starts always in the locked position. After this initial step, the servomotor only works when an unlock request is made through the smartphone application or by performing an automatic lock action as described in 3.19.

The most simple component present in the smart lock hardware is the green LED that has only the purpose to warn the user that is outside the door that, after an unlock request is made, the door is now unlocked and ready to open. The LED lights up every time the servo makes the rotation to the unlocked position. The LED is connected to the GND pin and to a digital pin as an output.

After every component is connected as shown in figure 4.5 and the hardware behavior is coded in the ESP8266 development board through Arduino IDE, the smart lock hardware should work as defined in 3.20.

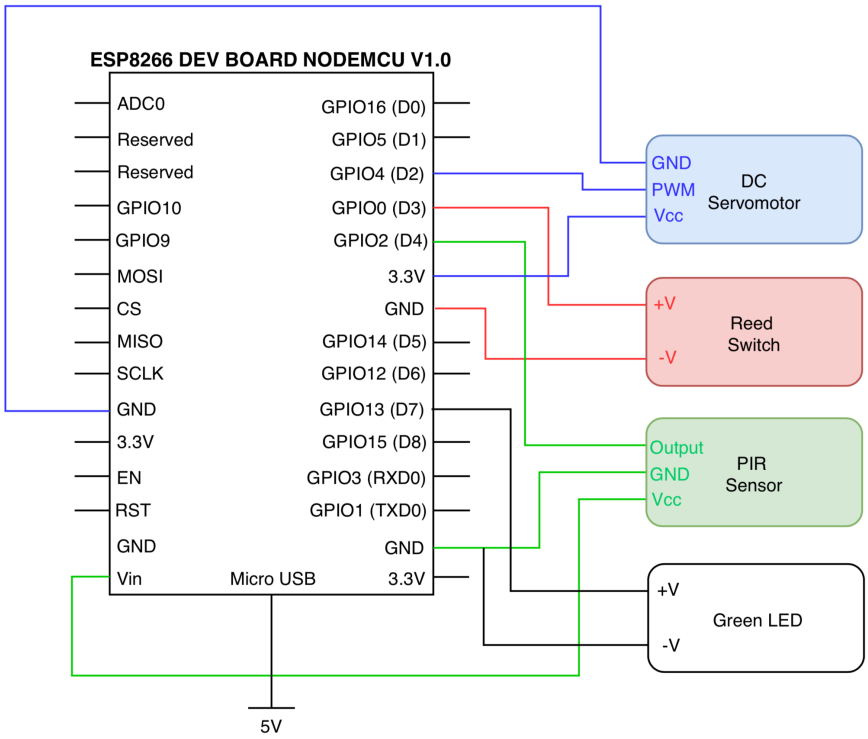


Figure 4.5: Smart lock hardware schematic

4.3 Smartphone application

On an initial stage of this work, the smartphone application was intended to be developed through Outsystems software, a Low-Code platform that enables a faster deployment of mobile and web applications [45], but due to unpredicted integration problems with the smart lock hardware, the smartphone application of this work was developed on Android Studio software.

The smartphone application development through Android Studio required Java programming, for the application's functionalities, and eXtensible Markup Language (XML) programming for the application's structure. To create a modern and user friendly layout for the application, Sketch [46] and Zeplin [47] softwares were also used.

4.3.1 Login

The log in functionality exists to create a security layer in accessing the smartphone application functionalities and also to identify each one of the users in order to understand each user's activity patterns to generate the cognitive notifications.

The screen in figure 4.6 is the first one that the user can interact with after opening the smartphone application in his own device.

After arriving at the login screen the user can insert the credentials, username and password, in the assigned input fields. For the login action to be triggered the user needs to press the button identified by "Sign In" for the system to be able to perform the user

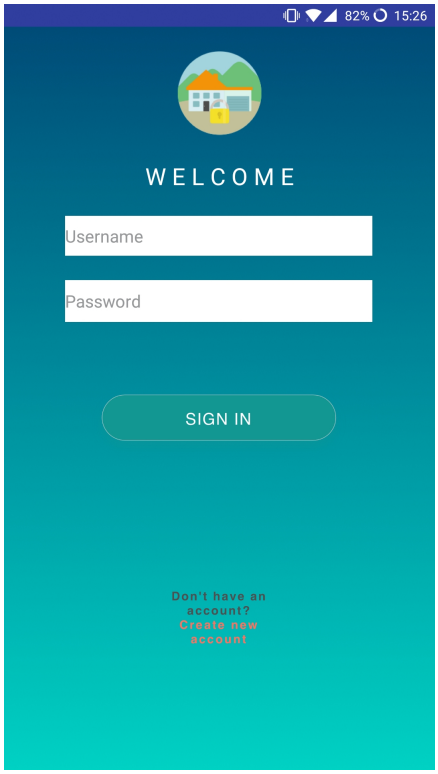


Figure 4.6: Smartphone application user login screen

credentials verification. When the sign in button is pressed, the smartphone application performs an Hypertext Transfer Protocol (HTTP) request to the cloud where it asks for an user credentials verification. This request can have two different possible outputs in the smartphone application, the credentials are valid and the user is now logged in or the credentials are not valid and the user sees an error message in his smartphone screen. When the user logs in, a new session is created in the cloud, every HTTP request that is made through that session is linked to that specific user through a unique session token, the session ends when the user logs out of the smartphone application.

If the user doesn't have a registered account, it can access to the the register screen through the login screen. There, the user is able to fill a register form in order to create an user account.

4.3.2 Register

To use all the smartphone application and system functionalities it's necessary to be a registered user. In the smartphone application login screen, as seen in figure 4.6, there is an option for new users to register themselves. By pressing the text "Don't have an account? Create new account" the smartphone application leads the user to the user register screen as shown in figure 4.7.

In the register screen the new user has to fill four different fields, username, firstname, lastname and password, and press the button identified by "Create New Account" text.

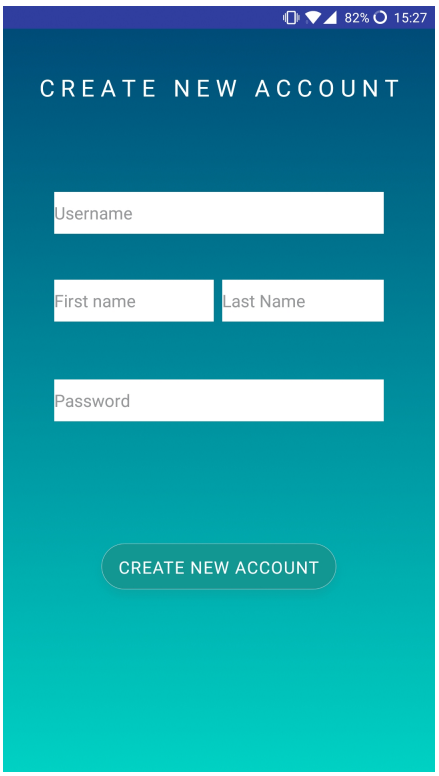


Figure 4.7: Smartphone application user register screen

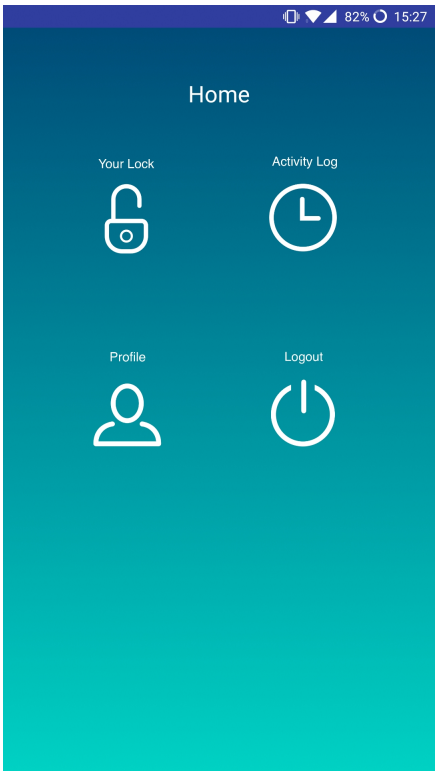
After pressing the button, the smartphone application verifies if the user inserted valid data, in this case it only verifies if the password has more than 6 characters and makes a register HTTP request to the cloud where the existence of the inserted username is checked. After this action there are two possible outcomes, the user sees a message that his register request is pending approval, or an error message indicating to check the inserted data.

When the registration is made, the user should wait for the system administrator approval. Altough this is not a smartphone application feature, the system administrator has to activate the user account through an HTTP request. That request is made through the Postman [48] software with the username as an input in order to activate the correct account, after this action the user is now registered and able to perform a login action that will lead him to all the smartphone application functionalities.

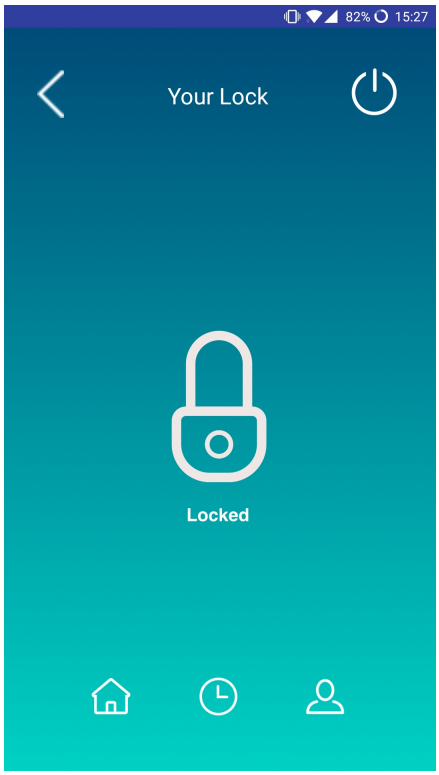
4.3.3 Logout

To create an extra layer of security, the smartphone application has a logout functionality. This functionality prevents that a strange person to the system could access the smartphone application functionalities. The logout action can be performed from several different screens. In figure 4.8 it's possible to see the logout button in the "Home" screen and in the right top corner of the "Your Lock" screen.

When a user presses one of the several logout buttons, the smartphone application



a Smartphone application Home screen



b Smartphone application Your Lock screen

Figure 4.8: Logout button in two different smartphone application screens

turns off the user access to the features and returns to the Login Screen. There is no need to end the user session in the cloud, since the next time he performs a login action a new session token will be assigned.

4.3.4 User profile functionality

Through the smartphone application the user is also able to see and update their own personal information. By accessing the "Profile" screen the user is able to see his own username, firstname, lastname fields and also a password field that can be filled with the new desired account password as shown in figure 4.10.

If the user wants to update one, or all, of the fields there is a need to press the field to edit, and after editing the user must press the "Save" button so that the smartphone application can make an HTTP request to update the user data in the cloud database. If the username doesn't exist in the database and the password meets the six character criteria, the user profile is updated. If not, an error message appears in the screen warning the user to check the edited data.

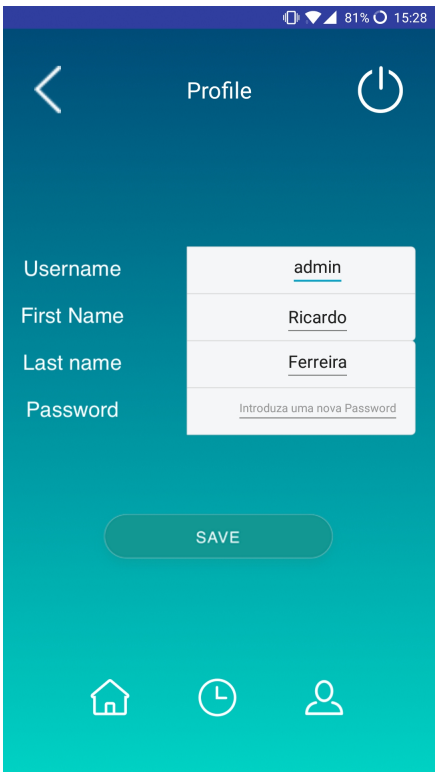


Figure 4.9: Smartphone application Profile screen

4.3.5 Activity log functionality

Smart lock system users can always check their own interactions with the smart lock through the "Activity Log" screen. Regular users can only see their own last ten activities, but the system administrator can see the last ten unlock action independently of the user who performed it. Both "Activity Log" screen scenarios are shown in figure **??**.

Every time a user goes to the "Activity Log" screen the smartphone application sends an HTTP request to update the activity data, to the cloud. The cloud runs a user verification in order to know if the user has administrator or regular user rights, this prevents that regular users can see other users activities by only allowing the system administrator to see that activity data. After this verification step, the cloud runs a database query and sends the updated activity log data to the user's smartphone application. Although this seems a time consuming process, it happens instantaneously after the user accesses the "Activity Log" screen.

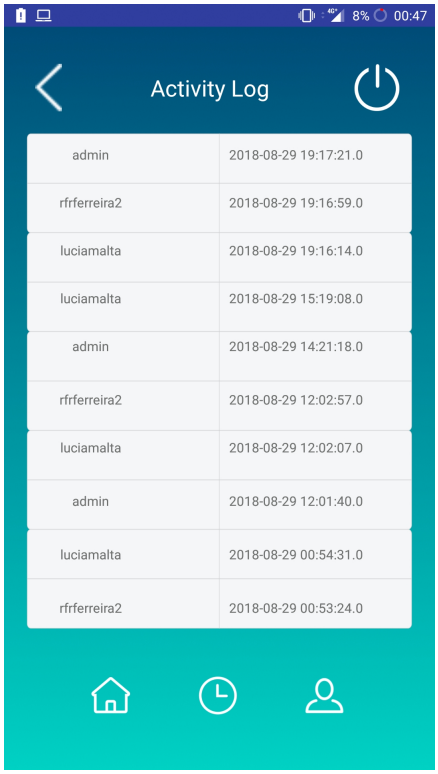


Figure 4.10: System administrator smartphone application Activity Log screen

4.3.6 Unlock

The unlock functionality is the one that allows the system user to interact, through an unlock action, with the smart lock hardware. This only happens through a successful integration between the system's three main modules, the smartphone application, the cloud platform and the smart lock hardware.

To unlock the smart lock, the logged in user has to be in the "Your Lock" screen. This screen is similar to the other functionalities screens and the big lock in the middle of the screen maked it very intuitive and user friendly. The screen is shown in figure 4.11.

For the smart lock to unlock, the user has to press the big lock in the center of the "Your Lock" screen. After the button is pressed an HTTP request is sent to the cloud that verifies the request by checking the user session token, then the cloud itself sends a request to the smart lock hardware that almost instantaneously performs the unlock action. The hardware performs other sensor readings in order to improve the security and the user experience as described in section 4.2.

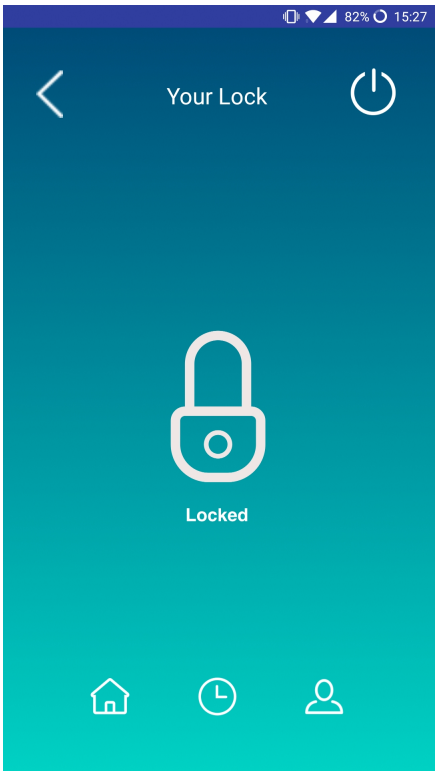


Figure 4.11: Smartphone application Your Lock screen

Chapter 5

Conclusions

This chapter presents and discusses this work's development and results. A future work approach is also made in order to help others to develop new Internet of Things (IoT), Home Automation and Smart Lock projects based in this one.

5.1 Results discussion

This work presents a smart lock system, based in Wi-Fi technology, that was built with the purpose to support a cognitive notification system that helps the system administrator to better manage his door lock activities and to have a broader view into the system user's routines. The whole system works reliably, providing the system's users a set of useful features that are in line with what is actually used in the smart lock products in the market.

This work adds a contribution to home automation and smart lock market by providing a cognitive notification system that relies on the user's activity database to predict their routines and to provide useful information to the system administrator. Despite the fact that this work only has one type of notification, that indicates when a specific user is late, it opens the possibility of building other types of notifications based on the system's usage acquired data. Other smart lock solutions that are already in the market only allow their system administrators to receive notifications that are based in real time events.

The communications that are held between the cloud platform and with the system's smartphone application and smart lock hardware run through Hyper Text Transfer Protocol (HTTP), and although all the HTTP requests are secured by OAuth 2.0, defined as the authorization standard nowadays, they are still vulnerable. This can be a real serious problem when talking about a system that can provide access to buildings, or even your own house. This indicates that the system communications need a clear security

improvement in order to protect the data that they carry.

The system was also designed with cloud platform operating as the core of it. It manages the system hardware and smartphone application requests, it stores the activity generated data and mainly it runs the cognitive notifications algorithm. Every communication has to cross the cloud is a way or another. The first system model was designed to have another communication flow, using Bluetooth Low Energy (BLE), between the smartphone application and the smart lock hardware in order to add another type of functionalities and security to the system. Unfortunately, due to hardware incompatibilities and time restrictions that communication flow was not implemented.

The smartphone application was intended to be developed in Outsystems software, but during the first prototype development, it showed a lot of communication incompatibilities with the remaining system, that delayed the overall work development. After this, the smartphone application development was made using Android Studio, Sketch and Zeplin softwares. This decision allowed the application to be user-friendly and clean, but at the same time, sharing the base functionalities that other, already in the market, smart lock systems have. Using Android Studio allowed that the developed smartphone application helped to create a personalized solution to control the smart lock system. The application proposed features work all the time, but some tweaks to the application must be made before it can be called a finished product. Sometimes when the session token expires and the user returns to the application, instead of showing the user the login screen, so that he can login again and receive a new session token, it just shows the user the home screen. Despite that fact, the user can't gain access to any of the available features, since the session token needed to perform the requests has already expired, but it's certainly a problem that needs to be fixed to achieve an even better user experience. The conclusion is that to develop a smartphone application for a system with these specifications, Outsystems is not the most suitable solution. Android Studio provides a better a better framework which allows the implementation of the needed features.

System's hardware plays the smart lock role itself. The system works properly, it helps in registering the user's activities with the door position sensor, and adds a layer of security with the presence sensor. ESP8266 is, in fact, a very good microcontroller to prototype projects like this, it's cheap, it has a major online community that supports its development and it can be programmed through Arduino IDE which makes developing easier. Despite the fact that the smart lock hardware plays a major role in the system, in its current development stage, it is too simple to apply in a real door, but due to time restrictions, no further developments were made to the hardware.

5.2 Future work

This work has a lot of ways in which it can be improved. To start, the system requests should become safer by implementing the Hypertext Transfer Protocol Secure (HTTPS) with which all the system communications become encrypted.

After enabling a more secure communication between all the system modules, the focus should be on the notification algorithms. Using the already gathered data there are a lot of new possibilities for generating new sets of notifications by developing new algorithms that would return different outputs.

In order for the system administrator to control the actual and eventual new types of notifications, a new smartphone application screen can be developed. Through this new screen, the system administrator can select which type of notifications he wants to receive or even define rules for those notifications, such as, only receive user is late notifications when the user is late for more than one hour, instead of the predefined thirty minutes. This would allow the system administrator to gain control over his own system.

Smart lock hardware should also be improved so that it can be applied to a door and several people can benefit from using it. The updated hardware should have a servomotor with more torque making it capable of rotating a door lock, or the key, or another possibility such as using a magnetic lock.

The hardware power supply was always made through the regular 5 V using a Universal Serial Bus (USB) cable that was also used to flash the code to the microcontroller during the prototype development. Wires and cables in a door is not a thing that people are used to seeing, to avoid that, a power supply circuit should be developed. Smart locks usually use disposable batteries to power the hardware, but a lithium battery approach would take smart locks a step further. There are several possibilities that should be studied in order to assure a reliable system to charge a smart lock lithium battery, such as harvesting energy from the door motion [49] or developing a charging system inside the door frame that connects to the smart lock every time the door is closed. A lithium battery based system would increase the device's reliability.

The actual prototype can control the user's door entries but not the exits. This disables new features possibilities such as creating new types of notifications based on new data generated by the exit routines. To enable those new features an exit register can be implemented. This feature can be developed by adding a new presence sensor on the inside of the building that detects when a person goes from the inside to the outside and by sniffing, through BLE, the nearby devices so that the system can acknowledge who's left.

To turn the smart lock into a more versatile, user-friendly and safe product, BLE should be implemented. It would also give more possibilities to new features that already exist in other smart solutions as auto-unlock where users just need to get close to the door for it to unlock.

After prototyping the new smart lock solution, the circuit should be printed into a printed circuit board and the hardware algorithms optimized to achieve a better energetic performance.

Bibliography

[1] R. van der Meulen. *Gartner Says 8.4 Billion Connected "Things"Will Be in Use in 2017, Up 31 Percent From 2016*. 2017. URL: https://www.gartner.com/newsroom/id/3598917 (visited on 01/11/2018).

[2] M. Krell. *What's Driving All The Home Automation Growth?* 2015. URL: https://www.forbes.com/sites/sethporges/2016/08/31/what-you-need-to-know-before-buying-a-smart-lock/#3fa539c51c54 (visited on 01/11/2018).

[3] iControl Networks. *2015 State of the Smart Home Report | Icontrol Networks*. 2015. URL: https://pt.slideshare.net/iangertler/2015-state-of-the (visited on 01/11/2018).

[4] *Evolution On Lock: From Stick to Gate*. URL: https://www.forbes.com/sites/stephanrabimov/2017/12/27/evolution-on-lock-from-stick-to-gate/{\#}418950307cac (visited on 02/17/2018).

[5] M. Wolf. *Smart Locks Expected To Be $3.6 Billion Slice Of Smart Home Pie*. URL: https://www.forbes.com/sites/michaelwolf/2014/02/27/smart-locks-expected-to-be-3-6-billion-slice-of-smart-home-pie/{\#}6bbecbc76990 (visited on 02/14/2018).

[6] Pew Research Center. *Demographics of Mobile Device Ownership and Adoption in the United States | Pew Research Center*. URL: http://www.pewinternet.org/fact-sheet/mobile/{\#} (visited on 02/14/2018).

[7] Nuki Home Solutions. *Smart Lock - Keyless electronic door lock for smart access - Nuki*. URL: https://nuki.io/en/ (visited on 01/17/2018).

[8] Nuki Home Solutions. *The Nuki Encryption Concept*. 2015. URL: https://nuki.io/en/blog/nuki-encryption-concept/ (visited on 01/22/2018).

[9] IFTTT. *IFTTT helps your apps and devices work together - IFTTT*. URL: https://ifttt.com/ (visited on 01/22/2018).

[10] I. August. *August Smart Lock | Your Smart Home Starts at the Door.* URL: http://august.com/ (visited on 01/22/2018).

[11] M. Fuller, M. Jenkins, and K. Tjølsen. "Security Analysis of the August Smart Lock". In: (2017). URL: https://courses.csail.mit.edu/6.857/2017/project/3.pdf.

[12] C. Reinisch, W. Kastner, G. Neugschwandtner, and W. Granzer. "Wireless technologies in home and building automation". In: *Industrial Informatics, 2007 5th IEEE International Conference on*. Vol. 1. IEEE. 2007, pp. 93–98.

[13] J.-S. Lee, Y.-W. Su, and C.-C. Shen. "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi". In: *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2007, pp. 46–51. ISBN: 1-4244-0783-4. DOI: 10.1109/IECON.2007.4460126. URL: http://ieeexplore.ieee.org/document/4460126/.

[14] K. Shahzad and B. Oelmann. "A comparative study of in-sensor processing vs. raw data transmission using ZigBee, BLE and Wi-Fi for data intensive monitoring applications". In: *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*. IEEE, 2014, pp. 519–524. ISBN: 978-1-4799-5863-4. DOI: 10.1109/ISWCS.2014.6933409. URL: http://ieeexplore.ieee.org/document/6933409/.

[15] Sigfox. "The world's first cellular network operator dedicated to M2M and IoT An ultra-efficient technology at your service Connect all your machines and objects to the Internet SIGFOX technology in a nutshell". In: (). URL: http://www.iotglobalnetwork.com/public/files/company/453198f6cdbabf2ef5d31f2113d1ee1c.pdf.

[16] *Amazon Web Services (AWS) - Cloud Computing Services*. URL: https://aws.amazon.com/ (visited on 03/02/2018).

[17] *Easy IoT*. URL: https://iot-playground.com/ (visited on 03/02/2018).

[18] *Google Cloud Computing, Hosting Services & APIs | Google Cloud Platform*. URL: https://cloud.google.com/ (visited on 03/02/2018).

[19] *IBM Cloud*. URL: https://www.ibm.com/cloud/ (visited on 03/02/2018).

[20] *Plataforma de Informática na Cloud e Serviços do Microsoft Azure*. URL: https://azure.microsoft.com/pt-pt/ (visited on 03/02/2018).

[21] Analog Devices. *Small, Low Power, 3-Axis ±3 g Accelerometer ADXL335*. Tech. rep. URL: https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf.

[22] *Reed Sensors vs. Hall Effect Sensors - Standex Electronics*. URL: https://standexelectronics.com/resources/technical-library/technical-papers/reed-sensors-vs-hall-effect-sensors/ (visited on 03/03/2018).

[23] *Magnetic contact switch (door sensor)*. URL: https://www.adafruit.com/product/375 (visited on 03/03/2018).

[24] *What Is an Ultrasonic Motion Detector? (with picture)*. URL: https://www.wisegeek.com/what-is-an-ultrasonic-motion-detector.htm (visited on 02/22/2018).

[25] *What is a Microwave Motion Detector: A Quick Guide | Protect America*. URL: https://www.protectamerica.com/home-security-blog/tech-tips/what-is-a-microwave-motion-detector-a-quick-guide{_}20042 (visited on 02/22/2018).

[26] P. Zappi, E. Farella, and L. Benini. "Tracking Motion Direction and Distance With Pyroelectric IR Sensors". In: *IEEE Sensors Journal* 10.9 (2010), pp. 1486–1494. ISSN: 1530-437X. DOI: 10.1109/JSEN.2009.2039792.

[27] B. Song, H. Choi, and H. S. Lee. "Surveillance Tracking System Using Passive Infrared Motion Sensors in Wireless Sensor Network". In: (2008), pp. 1–5. ISSN: 1550-445X. DOI: 10.1109/ICOIN.2008.4472790.

[28] "ESP-12E WiFi Module". In: (). URL: http://www.kloppenborg.net/images/blog/esp8266/esp8266-esp12e-specs.pdf.

[29] *INTRODUCTION TO NodeMCU ESP8266 JULY 2017 DEVKIT v1.0*. Tech. rep. URL: www.einstronic.com.

[30] "ESP32 Datasheet Espressif Systems About This Guide Documentation Change Notification". In: (). URL: https://www.espressif.com/sites/default/files/documentation/esp32{_}datasheet{_}en.pdf.

[31] *Technical Specs - Bean LightBlue*. URL: https://punchthrough.com/bean/docs/guides/getting-started/tech-specs/ (visited on 03/03/2018).

[32] *UML basics: The sequence diagram*. URL: https://www.ibm.com/developerworks/rational/library/3101.html (visited on 08/16/2018).

[33] *Firebase*. URL: https://firebase.google.com/ (visited on 05/11/2018).

[34] *HTTP Methods GET vs POST*. URL: https://www.w3schools.com/tags/ref{_}httpmethods.asp (visited on 07/11/2018).

[35] *Token Request - OAuth 2.0 Servers*. URL: https://www.oauth.com/oauth2-servers/device-flow/token-request/ (visited on 07/20/2018).

[36] *OAuth 2.0 — OAuth*. URL: https://oauth.net/2/ (visited on 07/20/2018).

[37] "RFC 6749 - The OAuth 2.0 Authorization Framework". In: (2012). ISSN: 2070-1721. URL: http://www.rfc-editor.org/info/rfc6749..

[38] *ESP8266 WiFi library*. URL: https://github.com/esp8266/Arduino/blob/master/libraries/ESP8266WiFi/src/ESP8266WiFi.h (visited on 07/10/2018).

[39] *PIR Motion Module HC-SR505*. Tech. rep. URL: www.rapidonline.com.

[40] S. Lee, K. N. Ha, and K. C. Lee. "A pyroelectric infrared sensor-based indoor location-aware system for the smart home". In: *IEEE Transactions on Consumer Electronics* 52.4 (2006), pp. 1311–1317. ISSN: 0098-3063. DOI: 10.1109/TCE.2006.273150.

[41] *Reed Switch Hookup Guide - learn.sparkfun.com*. URL: https://learn.sparkfun.com/tutorials/reed-switch-hookup-guide (visited on 07/13/2018).

[42] *Pull-up Resistors - learn.sparkfun.com*. URL: https://learn.sparkfun.com/tutorials/pull-up-resistors (visited on 07/13/2018).

[43] *Servo Motor SG-90 Basics, Pinout, Wire Description, Datasheet*. URL: https://components101.com/servo-motor-basics-pinout-datasheet (visited on 08/07/2018).

[44] *Servo library*. URL: https://www.arduino.cc/en/Reference/Servo (visited on 07/10/2018).

[45] *The #1 Low-Code Platform for Digital Transformation | OutSystems | OutSystems*. URL: https://www.outsystems.com/ (visited on 03/26/2018).

[46] *Sketch - The digital design toolkit*. URL: https://www.sketchapp.com/ (visited on 05/27/2018).

[47] *Zeplin*. URL: https://zeplin.io/ (visited on 05/27/2018).

[48] *Postman | API Development Environment*. URL: https://www.getpostman.com/ (visited on 06/28/2018).

[49] D. H. Litwhiler and T. H. Gavigan Penn State. *A Door Motion Energy Harvesting System for Powering an Electronic Door Lock*. Tech. rep., pp. 978–979. URL: http://cd14.ijme.us/papers/013{_}{_}DaleH.Litwhiler{\%}2CThomasH.Gavigan.pdf.