**DIGITALKEY – SIMULAREA COMPORTĂRII SENZORULUI DIN MÂNERUL UȘII MAȘINII**

**Candidat: Paula-Kristine Senciuc**

**Coordonator științific: SL dr.ing. Daniela Stănescu**

Sesiunea: Iunie 2023

**REZUMAT**

În cadrul lucrării s-au realizat două aplicații mobile specifice pentru sistemul de operare Android: una destinată utilizării de către administrator, iar una destinată clienților aplicației. În plus față de partea software, s-a realizat și o partea hardware printr-o aplicație Arduino care simluează prezența unei mașini. Comunicarea dintre cele două aplicații mobile și aplicația Arduino se realizează printr-un protocol de comunicare bazat pe tehnologia Bluetooth. Scopul lucrării este acela de a înlocui cheia fizică a unei mașini prin folosirea uneia digitale cu funcționalitățiile de închidere și deschidere a mașinii prezente în aplicația mobile a clientului. Pentru a putea observa starea în care se află mașina: deschisă sau închisă, s-a utilizat un led conectat la aplicația Arduino. Prin faptul că led-ul este apris se înțelege că mașina este încuiată, iar prin faptul că led-ul este stins se înțelege că mașina este descuiată. Comunicarea dintre Android și Arduino se realizeză în ambele sensuri. Prin apăsarea unui buton cu funcționalitatea ori de închidere, ori de deschidere a mașinii, se trimite un mescaj către Arduino, iar led-ul va arăta starea solicitată. În direcția opusă, aplicația Arduino prezintă un senzor tactil, iar când acesta este apăsat va trimite un mesaj înapoi către Android care va arăta cererea printr-o imagine și prin schimbarea functionalității butonului de deschidere/închidere.

Pentru a exdine folosirea aplicației, iar utilizarea ei să poată fi realizată pentru orice mașină, s-a adăugat o bază de date gestionată de către un administrator. La solicitarea aplicației de către un client, administratorul, folosind aplicația mobile destinată, va creea un cont ce prevede email, parolă, marca mașinii și adresa MAC a modulului bluetooth a mașinii. În cadrul aplicației administratorului, acesta poate vedea datele tuturor clienților și poate adăuga noi adrese MAC, care reprezintă noi mașini, fiecărui utilizator. Astfel, daca un utilizator dorește să folosească aplicația pentru mai multe mașini, este posibil.

În concluzie, lucarea simulează atât o cheie inteligentă prin aplicația mobile, cât și prezența unei mașini prin aplicația Arduino având un modul bluetooth cu o adresa unică MAC, ceea ce duce la atingerea scopului de a realiza o comunicare între cele menționate și a face posibile funcționalitățile de închidere și deschidere a unei mașini.

**CUPRINS**

[1. INTRODUCERE 4](#_Toc135068942)

[1.1 MOTIVAȚIA ALEGERII TEMEI 4](#_Toc135068943)

[1.2 OBIECTIVELE GENERALE ALE LUCRĂRII 4](#_Toc135068944)

[2. ANALIZA STADIULUI ACTUAL ÎN DOMENIUL CHEILOR DIGITALE 5](#_Toc135068945)

[3. FUNDAMENTARE TEORETICĂ 6](#_Toc135068946)

[3.1. LIMBAJE ȘI MEDII DE PROGRAMARE 6](#_Toc135068947)

[3.1.1 ANDROID 6](#_Toc135068948)

[3.1.2. ANDROID STUDIO 9](#_Toc135068949)

[3.1.3. JAVA 10](#_Toc135068950)

[3.1.4. FIREBASE 10](#_Toc135068951)

[3.1.5. ARDUINO IDE ȘI LIMBAJUL C 10](#_Toc135068952)

[3.2. COMPONENTE HARDWARE 11](#_Toc135068953)

[3.2.1. ARDUINO MEGA 2560 11](#_Toc135068954)

[3.2.2. MODULUL BLUETOOTH HC-06 12](#_Toc135068955)

[3.2.3. SENZORUL CAPACITIV TTP223B 13](#_Toc135068956)

[4. METODA DE PROIECTARE/DEZVOLTARE 14](#_Toc135068957)

[4.1. ARHITECTURA GENERALĂ 14](#_Toc135068958)

[4.2. APLICAȚIA ARDUINO 15](#_Toc135068959)

[4.2.1. CONTROLUL SENZORULUI TACTIL 16](#_Toc135068960)

[4.2.2. CONEXIUNEA CU APLICAȚIA ANDROID 17](#_Toc135068961)

[4.2.3. CONTROLUL LED-ULUI 17](#_Toc135068962)

[4.3. APLICAȚIA ANDROID PENTRU ADMINISTRATOR 17](#_Toc135068963)

[4.3.1. MODULARIZAREA APLICAȚIEI PENTRU ADMINISTRATOR 17](#_Toc135068964)

[4.3.1.1. MODULUL DE AUTENTIFICARE 17](#_Toc135068965)

[4.3.1.2. MODULUL DE ÎNREGISTRARE A UNUI CLIENT 18](#_Toc135068966)

[4.3.1.3. MODULUL DE ADĂUGARE DE NOI MAȘINI PENTRU CLIENȚI 18](#_Toc135068967)

[4.3.2. APLICAȚIA ANDROID PENTRU CLIENT 18](#_Toc135068968)

[4.3.2.1. MODULUL DE AUTENTIFICARE 18](#_Toc135068969)

[4.3.2.2. MODULUL DE PORNIRE ȘI OPRIRE A BLUETOOTH-ULUI 18](#_Toc135068970)

[4.3.2.3. MODULUL DE CĂUTARE ȘI CONECTARE A MAȘINII 18](#_Toc135068971)

[4.3.2.4. MODULUL DE ÎNCHIDERE ȘI DESCHIDERE A MAȘINII 18](#_Toc135068972)

[4.3.2.5. MODULUL DE DECONECTARE A APLICAȚIEI DE LA MASINĂ 18](#_Toc135068973)

[7. BIBLIOGRAFIE 19](#_Toc135068974)

# INTRODUCERE

## MOTIVAȚIA ALEGERII TEMEI

Lumea de astăzi e într-o dezvoltare continuă și extrem de rapidă, fiecare domeniu având parte de inovații din punct de vedere tehnologic chiar și de multiple ori pe an. Astfel oamenii nu doar se aliniază cu noua tehnologie pe care sunt nevoiți să o folosească la locurile de muncă, ci majoritatea sunt tot mai atrași de folosirea acesteia și în viata de zi cu zi. În anul 2023 tot mai mulți pasionați de tehnologie își doresc conectarea dispovitivelor pe care le dețin, de la computere, la weareble devices, la electrocasnice, la televizoarele smart până chiar și la mașină pentru a folosi toate acestea intr-un „ecositem” interconectat, smartphone-ul personal fiind mecanismul de control. Datorită companiei în care activez, Continental Automotive, unde contribui la dezvoltarea aplicațiilor în domeniul automotive, am avut o mare vizibilitate asupra dezvoltării tehnologice a unui produs pe care oamenii îl folosesc în general în viața de zi cu zi, mașina. La fel ca și smartphone-urile și mașinile au tot mai multe funcții, tot mai inovative de la un model la altul pentru a crea clientului o cât mai bună experiență cu produsul. Astfel, sub îndrumarea mentorului meu din cadrul companiei, am decis să dezvolt o aplicație care există în acest moment doar pentru unele mașini dar cel mai probabil va fi prezentă pe majoritatea mașiniilor în viitor – folosirea telefonului pentru a gestiona sistemul de închidere și deschidere al mașinii.

## OBIECTIVELE GENERALE ALE LUCRĂRII

# ANALIZA STADIULUI ACTUAL ÎN DOMENIUL CHEILOR DIGITALE

# FUNDAMENTARE TEORETICĂ

## 3.1. LIMBAJE ȘI MEDII DE PROGRAMARE

### 3.1.1 ANDROID

Android reprezintă un sistem de operare destinat în primul rând pentru dispozitive cu ecran tactil, cum ar fi telefoanele și tabletele.Acesta este dezvoltat în prezent de Google și se bazează pe un nucleu Linux [[1]](#one). Prin intermediul acestui nucleu care se află la baza stivei software, sistemul de operare Android interacționează cu părțile hardware ale unui dispozitiv [2].

O aplicație Android conține mai multe componente, iar cele de bază folosite pentru implementarea aplicației sunt Activity și Broadcast Receiver.

**Activity** este o clasă care reprezintă pentru utilizator un ecran cu care acesta poate intra în contact [3]. În figura 3.1 este caracterizată interacțuinea activităților, asftfel la pasul întâi activitatea unu este în prim plan, iar utilizatorul poate începe folosirea ei. La pasul al doilea se pornește a doua activitate, iar prima va fi pusă în stivă. La pasul al treilea începe cea de a treia activitate, iar cele două precedente sunt puse în stivă. În momentul în care se navighează înapoi de la activitatea trei la activitatea doi, activitatea trei va fi distrusă și activitatea doi va fi reluată.

Graphical user interface

Description automatically generated

Figura 3.1 – Interacțiunea activităților unei aplicații Android [4]

Fiecare activitate are propriul ei ciclu de viață care descrie evenimentele apărute în activitatea respectivă, iar fiecare eveniment este caracterizat de o stare [5].

|  |  |
| --- | --- |
| Eventiment | Stare |
| OnCreate() | se apelează în momentul în care activitatea este creată |
| OnStart() | se apelează în momentul în care activitatea este vizibilă utilizatorului aplicației |
| OnResume() | se apelează în momentul în care utilizatorul începe să interacționeze cu activitatea |
| OnPause() | se apelează în momentul în care activitatea care se desfășoară în momentul curent este pusă pe pauză și reluată activitarea precedentă |
| OnStop() | se apelează în momentul în care activitatea nu mai este vizibilă utilizatorului aplicației |
| OnDestroy() | se apelează înainte ca activitatea să fie distrusă de către sistem, fie manual, fie de sistem pentru a economisi memorie |
| OnRestart() | se apelează după ce activitatea a fost oprită și repornită |

Tabel 3.1 – Evenimentele împreună cu stările caracteristice prezente în ciclul de viață [5]

Diagram

Description automatically generated

Figura 3.2 – Ciclul de viață al unei activități al aplicației Android [5]

**Broadcast Receiver** este o componentă care descrie funcționalitatea aplicației

Android de a răspunde la mesaje provenite intern sau de la alte aplicații [6]. Există două tipuri pe care le poate avea componenta:

* Static Broadcast Receivers

Care se pot folosi nu doar când aplicația este pornită, ci și când este oprită, iar declararea acestora este prezentă în partea de fișier manifest.

* Dynamic Broadcast Receivers

Care se pot folosi doar în momentul în care aplicația este minimizată sau este în rulare de către utilizatorul acesteia [7].

În vederea oricărei aplicații Android se regăsește fișierul xml numit Manifest, creat automat la compilarea programului, care deține informații specifice și necesare fiecărei aplicații în parte. Acesta prezintă structura componentelor care se află în cadrul aplicației: fiecare Activity, Broadcast Receiver, Service și Provider, în funcție de existența implementării subclaselor ale componentelor menționate anterior. În plus, pentru fiecare componentă se precizează numele clasei Java, și modul de lansare a acestora [8].

Pentru a putea fi posibilă dezvoltarea aplicației, s-au folosit următoarele permisiuni aflate în interiorul fișierului Manifest.xml:

* BLUETOOTH\_CONNECT – folosit în cadrul aplicației pentru a putea opri și

porni bluetooth-ul telefonului mobil [9]

* BLUETOOTH\_ADMIN – folosit în cadrul aplicației pentru a putea căuta

dispozitive bluetooth în apropiere și pentru a se putea contecta la acestea [9]

* BLUETOOTH – folosit pentru a putea utiliza bluetooth-ul telefonului [9]
* BLUETOOTH\_SCAN – folosit pentru a putea dispune de funcționalitatea de a

căuta dispozitive în apropiere ce prezintă bluetooth-ul activat [9]

* ACCESS\_FINE\_LOCATION, ACCESS\_COARSE\_LOCATION and

*ACCESS\_BACKGROUND\_LOCATION* – folosite pentru a putea utiliza funcționalitatea de a căuta dispozitive bluetooth în apropiere [10]

* INTERNET și ACCESS\_NETWORK\_STATE – folosite pentru a putea utiliza

internetul pentru a avea acces la firebase

### 3.1.2. ANDROID STUDIO

În vederea dezvoltării aplicației Android s-a utilizat ca mediu de dezolvare (IDE – Integrated Development Environment) Android Studio, fiind cel oficial creat de Google. Acesta a fost conceput specific pentru Android, astfel prezintă un SDK – Software Development Kit gândit pentru a ajuta dezolvatorul pe tot parcursul aplicației. Acesta contribuie dezvoltării cu următoarele:

* Android SDK Build Tools – intervine în momentul procesului de compilare

convertind codul scris de către developer în cod mașină pentru a putea fi rulat pe un dispozitiv Android

* Android Virtual Device Manager – intervine în momentul testării aplicației,

oferind posibilitatea dezvoltatorului de a crea dispozitive virtuale Android, emulatoare, pentru a putea observa comportamentul aplicației în funcție de dispozitiv

* Layout Editor – intervine în momentul creării interfeței grafice cu utilizatorul cu

funcționalități de drag-and-drop pentru a ușura munca făcând să nu mai fie nevoie de scris cod manual [11].

### 3.1.3. JAVA

Pentru implementarea codului aplicației s-a ales folosirea limbajului de programare Java. Avantajul folosirii acestui limbaj de programare reprezintă posibilitatea rulării codului Java pe orice dispozitiv care suportă JVM – Java Virtual Machine putând fii portat cu ușurință după compilare, codul java devine bytecode, cod mașină, iar bytecode-ul este interpretat pentru a putea fi folosit pe mai multe platforme.

### 3.1.4. FIREBASE

Firebase reprezintă o colecție de servicii backend, dezvoltată de Google, care salvează și actualizează date pe cloud. Pentru o extindere a dezvoltării aplicației în care fiecare client poate avea propria adresă MAC a mașinii cu posibilitatea de adăugare a mai multor adrese MAC, s-a ales salvarea și actualizarea datelor într-o bază de date Firebase Realtime Database, astfel aplicația poate fi folosită de către mai mulți utilizatori. Prin intermediul acesteia data se salvează ca JSON [12].

Un alt beneficiu pe care Firebase îl aduce în dezvoltarea aplicației este observat prin intermediul serviciului: Firebase Authentication care oferă servicii de backend și este ușor de folosit având metode predefinite pentru crearea contului utilizatorului în care doar trebuie trimise ca parametrii emailul și parola, in cazul aplicației în cauză [12].

### 3.1.5. ARDUINO IDE ȘI LIMBAJUL C

Partea hardware a aplicației s-a implementat cu ajutorul mediului de dezvoltare Arduino IDE în care structura programului diferă față de programarea în limbajul C clasică, iar după procesul de compilare, cu ajutorul portului USB, programul va fi încarcat pe placa Arduino Mega 2560 [13].

Strunctura acestui program este împărțită în două secțiuni:

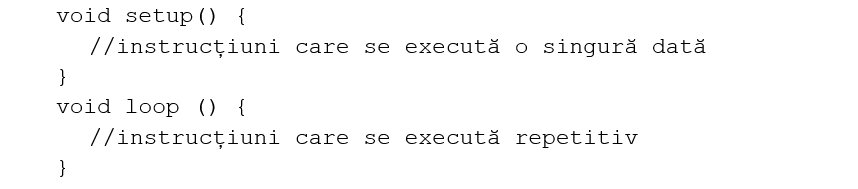


Figura 3.3 – Structura unui Program în Arduino IDE [13]

Secțiunea setup este executată o singură dată în momentul alimentării plăcii Arduino sau după ce butonul de Reset al plăcii este acționat. În această secțiune se inițializează variabilele utilizate în program, modul de lucru al terminalelor și se activează utilizarea bibliotecilor [13].

În cadrul secțiunii de loop, codul programului va fi executat ciclic atâta timp cât placa Arduino este contectată la o sursă de alimentare. Această secțiune descrie aplicația propriu-zisă [13].

Având în vedere specificațiile structurii unui program C în Arduino IDE se pot observa diferențele față de un program clasic C în care funcția clasică main s-a înlocuit cu cele două funcții: setup și loop

.

## 3.2. COMPONENTE HARDWARE

### 3.2.1. ARDUINO MEGA 2560

Sistemul hardware existent în aplicație prezintă ca și „creier al mașinii” placa de dezoltare Arduino Mega 2560 bazată pe microcontrolerul Atmega2560 care poate fi programată prin intermediul Arduino IDE [14].

**Specificații tehnice:**

Microcontroler: Atmega2560

Tensiunea de lucru: 5V

Tensiunea de intrare recomandată: 7-12V

Tensiuea de intrare (limită): 6-20V

Pini de intrare/ieșire digitali: 54 dintre care 15 pot fi folosiți ca PWM output

Pini de intrare analogici: 16

Curent per pin de intrare/ieșire: 20mA

Curent pentru 3.3V pin: 50mA

Memoria Flash: 256KB dintre care 8KB folosită pentru bootloader

SRAM: 8KB

EEPROM: 4KB

Clock Speed: 16 MHz

Led integrat: pinul 13 [14]

A close-up of a circuit board

Description automatically generated with medium confidence

Figura 3.4 – Arduino Mega 2560 [15]

### 3.2.2. MODULUL BLUETOOTH HC-06

Modulul Bluetooth HC-06 este folosit pentru comunicări cu transport de date fără fir pe o distanță relativ mică [16].

În contextul dezvoltării aplicației s-au folosit următorii pini:

|  |  |  |
| --- | --- | --- |
| Pinul | Numele | Funcționalitatea |
| 2 | Vcc | Are nevoie de tensiune de alimentare +5V pentru a alimenta modulul bluetooth |
| 3 | Gnd | Se conectează la masă |
| 4 | TXD | Datele se transmit serial de la modulul bluetooth prin intermediul acestul pin |
| 5 | RXD | Datele se primesc serial de la modulul bluetooth prin intermediul acestul pin |

Tabelul 3.2 – Descrierea Pinilor Modulului Bluetooth HC-06 [16]

Rolul acestui modul bluetooth în cadrul proiectului este de a realiza comunicarea dintre aplicația Android și aplicația Arduino, fiind o comunicație realizată pe bluetooth. Acest modul prezintă o adresă unică MAC care descrie unicitatea unei mașini, fiecare mașină are propria ei adresă MAC, iar prin intermediul acestei adrese MAC se poate realiza conexiunea între cele două aplicații, software și hardware.

Specificații tehnice:

Protocolul Bluetooth: Protocolul standard V2.0

Tensiunea de operare: 3.3 – 6V

Banda de frecvență: 2.40Ghz – 2.48 Ghz

Diagram, schematic

Description automatically generated

Figura 3.4 – Modulul Bluetooth HC-06 [17]

### 3.2.3. SENZORUL CAPACITIV TTP223B

Senzorul Capacitiv TTP223B funcționează cu o tensiune de alimentare cuprinsă intre 2,5V și 5V. Acesta are un timp de răspuns rapid de 60 ms, iar daca nu a mai fost atins de 12 s, va intra in modul de low power care va face ca timpul de răspuns să fie unul mai încet, mai exact de 220 ms [18].

În ceea ce privește utilizarea acestuia în cadrul aplicației dezvoltate, scopul este de a simula existența senzorului din mănerul ușii unei mașini, având și caracteristicile de deschidere și închidere observate prin intermediul stării unui led incorporat în placa Arduino și programat în Arduino IDE.

A picture containing text, electronics

Description automatically generated

Figura 3.5 - Senzor Capacitiv TTP223B [18]

# METODA DE PROIECTARE/DEZVOLTARE

## 4.1. ARHITECTURA GENERALĂ

Arhitectura generală este compusă din două aplicații Android care comunică cu două servicii: Firebase Authentication și Realtime Database Firebase. Aplicația Android destinată administratorului se ocupă cu gestionarea clienților, iar aplicația Android destinată utilizatorilor preia datele de logare, iar o data cu acestea câmpul specific adreselor MAC pentru a facilita comunicarea cu mașina.

În cadrul arhitecturii generale ale lucrării s-a adăugat și placa Arduino Mega 2560 care simulează existența unei mașini și comunica prin intermediul Bluetooth cu aplicația Android a clientului.

A picture containing text, electronics, screenshot, mobile phone

Description automatically generated

Figura 4.1 - Arhitectura generală

## 4.2. APLICAȚIA ARDUINO

Aplicația Arduino a fost construită în jurul ideii de a reprezenta o mașină împreună cu posibilitatea acesteia de a comunica cu telefonul pentru a facilita funcționalitățile de închidere și deschidere și verificare a acestora. Comunicarea se realizează prin intermediul bluetooth-ului cu trimiterea mesajelor: GET\_STATUS, LOCK și UNLOCK de la Android la Arduino și trimiterea mesajelor: STATUS\_LOCK și STATUS\_UNLOCK de la Arduino la Android. Închiderea și deschiderea mașinii poate fi făcută în două moduri, de la telefon din aplicația Android și prin atingerea seznorului mânerului mașinii care actualizează funcționalitățile din aplicația Android.

A picture containing text, screenshot, diagram, font

Description automatically generated

Figura 4.2 - Schema logică de funcționare a aplicației Arduino

### 4.2.1. CONTROLUL SENZORULUI TACTIL

Senzorul tactil este introdus în sistemul hardware al aplicației pentru a simula comportamentul unui senzor ce aparține mânerului mașinii. Funcționalitatea acestuia este descrisă cu ajutorul unui led ce indică starea mașinii. În momenul apăsării senzorului tactil, dacă ledul este aprins, ledul se va stinge și se va trimite către aplicația Android un mesaj ce indică noua stare, ledul stins, care reprezintă mașina în starea deschisă. Dacă ledul este stins, iar senzorul se apasă, starea acestuia se va schimba în aprins și aplicația mobile va fi înștiințată.

### 4.2.2. CONEXIUNEA CU APLICAȚIA ANDROID

Comunicarea între cele două aplicații prezente: aplicația Android și aplicația Arduino se realizează prin protocolul bluetooth al modulului HC06. Datele se trimit și se primesc serial, bit cu bit, mesajele transmise fiind pe 8 biți.

### 4.2.3. CONTROLUL LED-ULUI

Led-ul reprezintă mecanismul vizual care arată starea pe care o are mașina în funcție de comanda cerută, de închidere sau de deschidere. Acesta este controlat atât din aplicația Android destinată clientului, cât și din aplicația Arduino.

## 4.3. APLICAȚIA ANDROID PENTRU ADMINISTRATOR

### 4.3.1. MODULARIZAREA APLICAȚIEI PENTRU ADMINISTRATOR

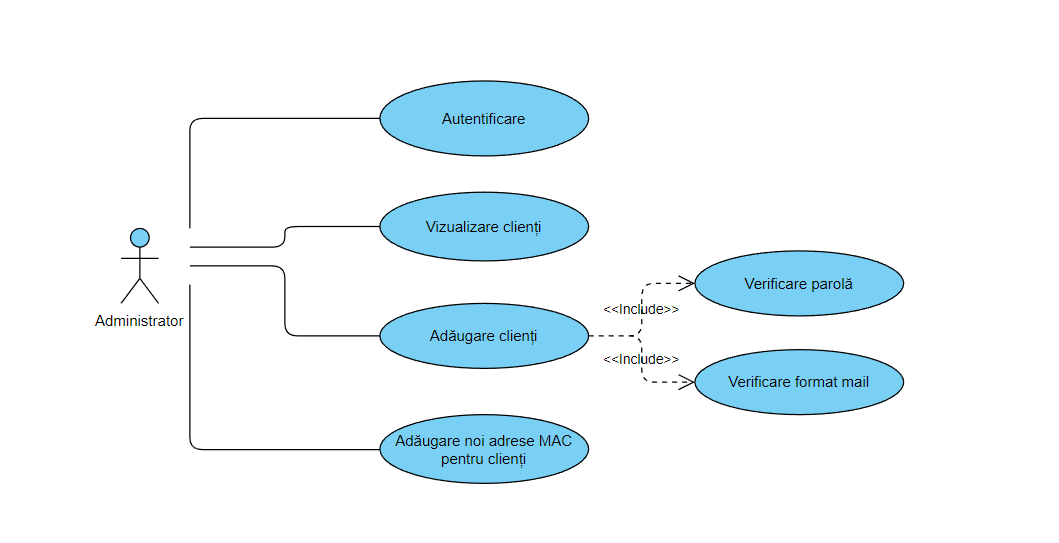


Figura 4.3 Diagrama use case pentru aplicația administratorului

#### 4.3.1.1. MODULUL DE AUTENTIFICARE

Modulul de autentificare conține două câmpuri: adresa de mail și parola. În cadrul acestui proiect s-a ales să nu existe funcționalitatea de înregistrare pentru administrator. În cazul administratorului, el primește un cont destinat creat din baza de date, iar autentificarea se va realiza cu datele primite.

#### 4.3.1.2. MODULUL DE ÎNREGISTRARE A UNUI CLIENT

Administratorul are responsabilitatea ca la cererea creării unui cont pentru un nou client, acesta să adauge în baza de date datele noului client: adresa de mail, parola și câmpul destinat adreselor MAC (fiecare adresă este destinată pentru o mașină). În cazul în care parola introdusă în momentul creării continul conține mai putin de șapte caractere va apărea un mesaj de eroare, iar dacă adresa de mail aleasă pentru crearea contului nu are un format valid undei adrese de mail va fi afișat un mesaj corespunzător. La apariția celor două erori menționate anterior, înregistrarea unui nou client nu va fi efectuată.

#### 4.3.1.3. MODULUL DE ADĂUGARE DE NOI MAȘINI PENTRU CLIENȚI

Scopul funcționalității de a adăuga noi mașini pentru clienți vine din nevoia de a putea gestiona mai multe mașini folosind același cont de pe o singură aplicație din telefonul personal. Astfel un client care deține mai multe mașini, le poate controla pe toate folosind un sigur cont, fără a fi nevoit să se înregistreze cu credențiale diferite de fiecare dată. Administratorul va introduce atât adresa MAC necesară conectării la Arduino, cât și modelul mașinii.

### 4.3.2. APLICAȚIA ANDROID PENTRU CLIENT

#### 4.3.2.1. MODULUL DE AUTENTIFICARE

#### 4.3.2.2. MODULUL DE PORNIRE ȘI OPRIRE A BLUETOOTH-ULUI

#### 4.3.2.3. MODULUL DE CĂUTARE ȘI CONECTARE A MAȘINII

#### 4.3.2.4. MODULUL DE ÎNCHIDERE ȘI DESCHIDERE A MAȘINII

#### 4.3.2.5. MODULUL DE DECONECTARE A APLICAȚIEI DE LA MASINĂ

# 7. BIBLIOGRAFIE

[1] Aliferi, C., *Android Programming Cookbook,* 2016.

[2] Singh R., *An Ovewview of Android Operating System and Its Security Features,* 2014.

[3] \*\*\* <https://data-flair.training/blogs/android-application-components/> accesare 11 mai 2023)

[4] \*\*\* <https://developer.android.com/guide/components/activities/tasks-and-back-stack accesare 11 mai 2023>)

[5] \*\*\* <https://www.geeksforgeeks.org/lifecycle-in-android-architecture-components/> accesare 11 mai 2023)

[6] Walls, C., *Open Source, Embedded Linux, and Android,* Embedded Software, 2012.

[7] \*\*\* <https://www.geeksforgeeks.org/broadcast-receiver-in-android-with-example/> accesare 11 mai 2023)

[8] \*\*\* [https://emteria.com/learn/android-manifest-file](https://emteria.com/learn/android-manifest-file accesare 11 mai 2023)  accesare 11 mai 2023)

[9] \*\*\* <https://developer.android.com/reference/android/Manifest.permission>

accesare 11 mai 2023)

[10] \*\*\* <https://developer.android.com/reference/android/bluetooth/BluetoothAdapter> accesare mai 2023)

[11] \*\*\* <https://www.pangea.ai/dev-mobile-app-resources/the-pros-and-cons-of-android-studio-and-app-tools/> accesare 11 mai 2023)

[12] \*\*\* <http://firebase.google.com/> accesare 12 mai 2023)

[13] M. Popa, „Curs Sisteme Încorporate”, Universitatea Politehnica Timișoara, 2022.

[14] \*\*\* [https://docs.rs-online.com/6560/A700000007742264.pdf](https://docs.rs-online.com/6560/A700000007742264.pdf%20accesare%2011%20mai%202023)  accesare 11 mai 2023)

[15] \*\*\* [https://www.amazon.de/Arduino-Mega-2560-R3-Microcontroller/dp/B0046AMGW0 accesare](https://www.amazon.de/Arduino-Mega-2560-R3-Microcontroller/dp/B0046AMGW0 accesare 11 mai 2023)  accesare 11 mai 2023)

[16] \*\*\* https://components101.com/wireless/hc-06-bluetooth-module-pinout-datasheet accesare 11 mai 2023)

[17]\*\*\*<https://classes.engineering.wustl.edu/ese205/core/index.php?title=Bluetooth_Module_%28HC-06%29_%2B_Arduino> accesare 11 mai 2023)

[18] \*\*\* https://robotica.md/senzor-capacitiv-ttp223b accesare 11 mai 2023)