

Facultatea de Automatică și Calculatoare

Programul de Master: Sisteme Informatice Aplicate în Producţie şi Servicii



**Sistem de pontare bazat pe identificare biometrică prin amprentă digitală**

**Teză de disertaţie**

Conducător științific:

Conf.dr.ing. Florin DRĂGAN

Autori:

Oana Maria PALCU

Vlad Alexandru MEICI

poză

Timişoara,

2018

Cuprins

[1. Introducere. Punerea problemei 3](#_Toc514792588)

[1.1. Conceptul 3](#_Toc514792589)

[1.2. Generalități 3](#_Toc514792590)

[1.2.1. Dispozitivul de pontare 3](#_Toc514792591)

[1.2.2. Aplicația web 4](#_Toc514792592)

[2. Analiza stadiului actual în domeniul problemei 4](#_Toc514792593)

[2.1. Stadiul actual 4](#_Toc514792594)

[2.2. Dispozitive asemănătoare 6](#_Toc514792595)

[2.2.1. FPC-4500 6](#_Toc514792596)

[2.2.2. FPA-1700 6](#_Toc514792597)

[2.2.3. FPA-1600 7](#_Toc514792598)

[2.2.4. PRO-CAPTURE-T 8](#_Toc514792599)

[2.2.5. SilkFP-101TA 9](#_Toc514792600)

[3. Bazele teoretice 10](#_Toc514792601)

[3.1. Identificare biometrică 10](#_Toc514792602)

[3.2. Dactiloscopie 12](#_Toc514792603)

[3.3. Conexiune Wi-Fi 13](#_Toc514792604)

[3.4. Programare orientată pe obiecte 15](#_Toc514792605)

[3.5. Principiile SOLID 16](#_Toc514792606)

[3.6. Modelul arhitectural MVC 17](#_Toc514792607)

[3.7. Dezvoltarea aplicațiilor web 18](#_Toc514792608)

[3.8. Controlul versiunilor 19](#_Toc514792609)

[3.9. Imprimare 3D 20](#_Toc514792610)

[4. Soluţia propusă şi metodologia de proiectare și dezvoltare 20](#_Toc514792611)

[4.1. Satisfacerea cerințelor de funcționare 20](#_Toc514792612)

[4.1.1. Dispozitivul de pontare 20](#_Toc514792613)

[4.1.2. Aplicația web 21](#_Toc514792614)

[4.2. Componente 21](#_Toc514792615)

[4.2.1. Arduino Mega 2560 21](#_Toc514792616)

[4.2.2. Senzor de amprentă 22](#_Toc514792617)

[4.2.3. 2.8" TFT Touchscreen Shield 24](#_Toc514792618)

[4.2.4. WiFi Shield 26](#_Toc514792619)

[4.2.5. Piezo 28](#_Toc514792620)

[4.2.6. Buton On/Off Switch 29](#_Toc514792621)

[4.2.7. Diverse 30](#_Toc514792622)

[4.3. Tehnologii folosite 30](#_Toc514792623)

[4.3.1. Programe software 30](#_Toc514792624)

[4.3.1.1. Arduino IDE 30](#_Toc514792625)

[4.3.1.2. Visual Studio 30](#_Toc514792626)

[4.3.1.3. Visual Studio Code 31](#_Toc514792627)

[4.3.1.4. SQL Server Management Studio 31](#_Toc514792628)

[4.3.1.5. Microsoft SQL Server 32](#_Toc514792629)

[4.3.1.6. Microsoft IIS 32](#_Toc514792630)

[4.3.1.7. Tinkercad 32](#_Toc514792631)

[4.3.2. Limbaje de programare 33](#_Toc514792632)

[4.3.2.1. C 33](#_Toc514792633)

[4.3.2.2. HTML 34](#_Toc514792634)

[4.3.2.3. CSS 35](#_Toc514792635)

[4.3.2.4. JavaScript 36](#_Toc514792636)

[4.3.2.5. C# 36](#_Toc514792637)

[4.3.2.6. SQL 37](#_Toc514792638)

[4.3.3. Tehnologii web 38](#_Toc514792639)

[4.3.3.1. ASP.NET 38](#_Toc514792640)

[4.3.3.2. Entity Framework 39](#_Toc514792641)

[4.3.3.3. Bootstrap 42](#_Toc514792642)

[4.3.3.4. jQuery 42](#_Toc514792643)

[4.3.3.5. Ajax 43](#_Toc514792644)

[4.3.3.6. LINQ 44](#_Toc514792645)

[5. Implementare 44](#_Toc514792646)

[5.1. Implementare hardware 44](#_Toc514792647)

[5.2. Implementare software 44](#_Toc514792648)

[5.3. Implementare firmware 44](#_Toc514792649)

[6. Interfața cu utilizatorul 44](#_Toc514792650)

[6.1. Dispozitivul de pontare 44](#_Toc514792651)

[6.2. Aplicatia web 44](#_Toc514792652)

[7. Concluzii, contribuţii şi direcţii de continuare a dezvoltării 45](#_Toc514792653)

[8. Bibliografie 45](#_Toc514792654)

# Introducere. Punerea problemei

## Conceptul

Nevoia pronunţată de a determina sau verifica identitatea persoanelor a dus la creşterea importanţei cercetării în domeniul autentificării biometrice. Autentificarea biometrică reprezintă ştiinţa stabilirii unei identităţi, bazându-se pe atributele fizice sau comportamentale ale unui individ, dintre acestea făcând parte amprenta, faţa, vocea, mersul, irisul, semnătura, geometria mâinii, urechea, etc.

Faptul că fiinţele umane au amprente şi acestea sunt diferite a fost cunoscut încă din preistorie. Există chiar nişte străvechi tăbliţe chineze pe care amprentele apar ca un mijloc de identificare a autorilor tăbliţelor.

Se poate spune, deci, că amprentele au fost folosite ca mijloc de identificare încă de acum 2000 de ani, însă există o mare diferenţă între a şti că oamenii au amprentele diferite şi a folosi ştiinţific acest lucru.

Când au fost dezvoltate primele tehnici biometrice, cu decenii în urmă, acestea erau prea scumpe şi complexe, fiind folosite numai la aplicaţii militare de înaltă securitate. Situaţia însă, s-a schimbat dramatic datorită progresului tehnologiei informatice şi, totodată, creşterii explozive a fraudelor.

## Generalități

### Dispozitivul de pontare

Dispozitivul de pontare are 3 stări:

* Scanare amprentă la sosire
* Scanare amprentă la plecare
* Scanare amprentă la înregistrare

Cele 3 stări sunt vizibile pe un touchscreen care dispune de o interfață simplă și prietenoasă cu mesaje de avertizare dacă scanarea a fost realizată cu succes și butoane pentru a defini starea scanării.

În modul de înregistrare, pe touchscreen se afișează mesaje de instrucțiuni cu privire la finalizarea înregistrarii în baza de date a unei noi persoane. Înregistrarea poate fi finalizată doar prin intermediul aplicației web pe baza unui cod PIN.

### Aplicația web

Aplicația web realizează umătoarele operații:

* Rapoarte de pontaj pentru angajații unei firme
* Operații de modificare, adăugare, ștergere asupra bazei de date
* Generare cerere de concediu legal
* Înregistrare concediu medical
* Calendar planning
* Autentificare utilizator

Rol utilizator:

* Administrator
* Utilizator

# Analiza stadiului actual în domeniul problemei

### Stadiul actual

Fiecare companie trebuie sa faca pontajul angajatilor, adica sa inregistreze orele pe care fiecare salariat le presteaza, pentru ca la sfarsitul lunii sa poata sa fie facuta plata salariului. Mai mult, legislatia in domeniul muncii s-a schimbat, iar fisa de pontaj a devenit obligatorie.

Fiecare angajator este obligat sa tina la sediul companiei evidenta orelor de munca ale fiecarui salariat in parte. Legea spune clar ca trebuie sa fe evidentiata ora de incepere a serviciului si ora la care s-a terminat programul de lucru. De asemenea, angajatorul este obligat sa arate documentele care tin aceasta evidenta inspectorilor teritoriali de munca ori de cate ori acestia solicita acest lucru. Angajatii nu au obligatia de a semna fisa de pontaj, ei pot face acest lucru numai in cazul in care Regulamentul intern al companiei unde lucreaza specifica acest lucru.

* Condica de prezenta, depasita de situatie?

Scopul masurii adoptate de Guvern este acela de a reduce munca la negru, dar odata cu aceasta noua regula, angajatorii au parte de mai multe responsabilitati. In contextul actual, condica de prezenta, in care evidenta angajatilor se tine pe hartie, pare acum o solutie greoaie, depasita, cu un risc enorm de greseala, care ingreuneaza munca departamentului de Resurse Umane sau a administratorului firmei, depinde de la caz la caz, de cine tine evidenta prezentei la munca a angajatilor.

In plus, un astfel de formular se poate deteriora, din diverse cauze. Este nevoie si de spatiu de depozitare pentru dosarele care se aduna de la o luna la alta. Mai mult, in cazul unui eveniment nefericit, precum un incendiu, se poate pierde definitiv.

Evidenta orelor de munca se mai tine si prin pontajul intocmit zi de zi, in baza unei fise colective de prezenta, de cele mai multe ori un document Excel. Insa si aceasta solutie pare depasita in contextul evolutiei actuale a tehnologiei.

* Sistem de pontaj, o solutie ideala.

Un sistem de pontaj pentru a usura munca departamentului HR, nu numai in ceea ce priveste pontajul zilnic al angajatilor, pentru a evidentia timpul lucrat pe ore, ci si pentru aprobarea cererilor de concediu, pentru a defini programe de lucru, pentru a tine evidenta angajatilor pe centre de cost si asa mai departe.

Un sistem de pontaj elimina erorile umane care pot aparea in cazul fisei clasice de pontaj ori in cazul utilizarii programului Excel.

Softul are sisteme integrate, ceea ce inseamna ca poate sa fie accesat de pe mai multe device-uri: tableta, PC sau smartphone. Asta inseamna ca angajatii si angajatorii pot sa il acceseze si de la distanta, ceea ce inseamna ca va creste productivitatea companiei. Angajatorul are acces in orice moment la informatii despre fiecare angajat in parte, nu mai pierde timp pretios cautand prin dosare.

In plus, un astfel de sistem poate sa genereze rapoarte de configurare atat in PDF, cat si in Excel.

* Sistemul permite introducerea datelor cu usurinta.

In etapa de implementare a unui astfel de sistem de pontaj, compania care il implementeaza va face o analiza a firmei unde urmeaza sa instaleze sistemul. In baza acesteia se fac configurarile dupa care, fiecare utilizator va beneficia de training, ca sa stie cum sa foloseasca sistemul. Oricum, sistemul de pontaj este usor de folosit, pentru ca este construit in asa fel incat sa permita introducerea cu usurinta a datelor. Sistemul poate fi usor inteles de catre fiecare angajat in parte, iar daca exista neclaritati, poate oricand sa ceara ajutorul companiei care l-a instalat.

In sistem exista reguli de validare predefinite pentru pontaj, in conformitate cu legislatia muncii, intotdeauna actualizate, dar exista si posibilitatea introducerii de reguli customizate.

(\*\*\*, 2018, )

### Dispozitive asemănătoare

#### FPC-4500



Colector de amprente USB

Se foloseste pentru colectarea amprentelor fara necesitatea deplasarii administratorului la aparatul de pontaj.

Amprentele colectate se distribuie aparatului de pontaj prin retea.

Caracteristici: rezolutie 500 DPI; suprafata scanata 14.2x18.4 mm, este compatibil cu USB 2.0

Senzor amprenta: optic

Dimensiuni: 65.5x49x79.8mm

(\*\*\*, 2018, )

#### FPA-1700



Controler de acces cu functie de pontaj, cu cititoare de amprente si de cartele incorporate

Rezistenta extrema la conditiile de mediu (IP65, pana la minus 20°C)

Cartele: 125KHz (EM4100 sau compatibil)

Alimentare: 12Vcc, 3A

Consum /Standby: 500mA, 50mA

Capacitate utilizatori: 10000 cartele si 1500 amprente

Evenimente stocabile: 30000

Conectivitate: TCP/IP, RS485, Wiegand 26/34 (2 porturi - intrare cititor extern, iesire pentru conectare in cadrul unui sistem integrat de control acces)

Control acces: senzor usa, alarma, buton iesire, sonerie, incuietoare electrica fail-safe sau fail-secure (releu)

Alte caracteristici: interfata si ghid vocal in limbile romana, maghiara si engleza, anti pass-back, anti-tamper

Dimensiuni: 185x62.5x41.5mm

Temperatura de functionare: -20-50C

NIvel de umiditate admis: 20-80%

(\*\*\*, 2018, )

#### FPA-1600



Controler de acces cu functie de pontaj, cu cititoare de amprente si decartele incorporate.

Rezistenta extrema la conditiile de mediu (IP65, pana la -40°C).

Cartele: 125KHz (EM4100 sau compatibil)

Alimentare: 12Vcc, 3AConsum / standby: 500mA / 50mA

Capacitate utilizatori: 10 000 cartele si 1 500 amprente

Evenimente stocabile: 30 000

Conectivitate: TCP/IP, RS485, Wiegand 26/34 (2 porturi - intrare cititor extern, iesire pentru conectare in cadrul unui sistem integrat de control acces)

Control acces: Senzor usa, alarma, buton iesire, sonerie, incuietoare electrica fail-safe sau fail-secure (releu)

Alte caracteristici: anti pass-back, anti-tamper

Dimensiuni: 185x62.5x41.5 mm

Temp. de functionare: -40 – 50°C (cu incalzitor inclus)

Nivel de umiditate admis: 20 – 80%

(\*\*\*, 2018, )

#### PRO-CAPTURE-T



Controler stand-alone cu functie de pontaj, cu amprenta, card si tastatura, ZK Bio Security

Mod de identificare: Amprente, carduri RFID, cod numeric (PIN)

Capacitate utilizatori (total): 10000

Capacitate carduri/PIN/Amprente/Fete/Vene: C: 10000/P: 10000/A: 6000/F: -/V:

Capacitate evenimente stocate: 100000

Carduri compatibile: EM 125kHz sau compatibil (MIFARE, DESFIRE optional)

Distanta citire: 2-8cm

Combinatie deschidere usa: Amprente, card, PIN (combinatii) / captura foto eveniment

Mod de programare: Software (ZKBioSecurity), tastatura touch

Sunet: Voce Hi-Fi

Ecran: LCD 2.4” color

Camera video: Camera Dual IR, 120° (H)

Comunicatie: TCP/IP, RS485 (cititor secundar amprente), SRB, USB host, WG26/34 in, WG26/34 out

Comanda incuietoare: Releu (NO/NC), mono-stabil temporizat, 1-255 secunde

Curent maxim incuietoare: 2A/30V

Conexiuni auxiliare: Senzor usa (NO/NC), intrare auxiliara (NO/NC), buton iesire (NO), iesire alarma (NO), iesire sonerie

Functii: Pontaj, anti-passback, nivele acces / grupuri / sarbatori, sonerie programabila, buton sonerie

Protectie: Anti-demontare, incercari acces esuate

Alimentare: 12Vcc / 500mA

Material carcasa: ABS

Dimensiuni: 195.5(H) x 87(l) x 35(A)mm

Masa bruta : 1.50kg

Temperatura / Umiditate: -10°C ~ +50°C, 20% ~ 80%

Aplicabilitate: Interior

(\*\*\*, 2018, )

#### SilkFP-101TA



Terminal de pontaj cu ecran TFT 3.5”, senzor amprenta SilkID. Cu un algoritm de verificare inovativ, senzorul de amprente SilkID incorporat realizeaza rapid si precis identificarea chiar si a amprentelor umede, uscate sau aspre. Tehnologia de detectie infrarosu permite detectia automata a amprentelor si auto-activarea senzorului pentru a economisi energia. Functioneaza in conditii de iluminare puternica datorita tehnologiei anti-glare. Este potrivit pentru institutii guvernamentale, financiare, fabrici, santiere.

Capacitate utilizatori: 10000

Capacitate amprente: 3000

Capacitate carduri: 10000

Capacitate evenimente: 100000

Senzor amprenta: SilkID

Ecran: TFT, 2.8” color

Comunicatie: TCP/IP, USB host/client

Functii standard: Alarme programabile, schimbare status programabila, cod operatie, SMS, DST Server Web, foto ID

Alimentare: 5Vcc / 0.8A

Temperatura: 0°C ~ +45°C

Umiditate: 20% ~ 80%

Dimensiuni: 192(L) x 140(H) x 42.5(A)mm

SDK: Profesional

(\*\*\*, 2018, )

# Bazele teoretice

## Identificare biometrică

Biometria, termen derivat din cuvintele greceşti bios (viaţă) şi metrikos (măsură), reprezintă un complex de metode automatizate destinate identificării unei persoane folosind unele caracteristici biometrice (geometria palmelor, amprenta digitală, irisul, retina, geometria feţei, greutatea corpului, presiunea sanguină, etc.) sau comportamentale (timbrul vocal, configuraţia ADN, dinamica scrisului, scanarea semnături, dinamica acţionării tastelor, etc.) ale acesteia, ştiut fiind faptul că unele dintre aceste caracteristici biometrice, de exemplu amprentele digitale sau irisul, ori comportamentale, de exemplu timbrul vocal, pot identifica în mod unic o persoană. Dată fiind această unicitate, informaţiile biometrice pot fi folosite pentru proiectarea şi implementarea unor tehnologii, echipamente şi sisteme destinate diseminării identităţii cu performanţe mult superioare celor existente.

Una dintre cele mai cunoscute caracteristici biometrice este reprezentată de amprenta digitală.

În termenii tehnologiei informaţiei, biometria este asociată tehnologiilor şi tehnicilor destinate securităţii şi confirmării identităţii pe baza unor caracteristici biologice individuale, măsurabile ale persoanei. De exemplu, amprentele digitale, geometria mâinii, geometria feţei, amprenta (codul) irisului sau a retinei, timbrul vocal pot fi utilizate în sisteme şi scheme destinate accesului la un computer, într-o anumită încăpere şi, de ce nu, la un cont bancar.

Identificarea automată a persoanei reprezintă procesul prin care un sistem biometric asociază o persoană unei identităţi specifice, acest lucru putându-se realiza în termenii verificării sau recunoaşterii. În cadru procesului de verificare, sistemul numai autentifică o identitate pretinsă; cu alte cuvinte sistemul verifică dacă o persoană este cine pretinde că este.

În cadrul recunoaşterii creşte complexitatea procesului, sistemul determinând identitatea persoanei prin consultarea unei baze de date sau prin testarea unei reţele neuronale cu informaţii despre persoane, cu alte cuvinte, sistemul determină cine este persoana respectivă fără ca aceasta să precizeze un nume ori alte informaţii de identificare.

Proiectarea unui sistem de verificare este mult mai simplă decât a unuia de recunoaştere a identităţii. Un sistem de verificare autentifică identitatea pretinsă de persoană prin compararea trăsăturilor biologice particulare furnizate de aceasta la un anumit moment cu măsurile acestor trăsături memorate anterior în sistem şi asociate identităţii pretinse de persoană; de exemplu, amprenta degetului inelar a unei persoane ce pretinde a fi Popescu este comparată cu amprenta degetului inelar a lui Popescu memorată anterior într-o bază de date sau într-o reţea neuronală a sistemului. Spre deosebire de sistemul de verificare, sistemul de recunoaştere are o complexitate procedurală mai crescută, trăsăturile biometrice furnizate de o persoană fiind comparate cu măsurile tuturor trăsăturilor biometrice similare stocate într-o bază de date sau într-o reţea neuronală. De exemplu, amprenta unei persoane ce doreşte accesul într-o cameră cu destinaţie specială este comparată cu amprentele tuturor persoanelor autorizate să intre în acea încăpere. Un alt exemplu concludent îl constituie accesul neautorizat într-un autoturism personal. Există sisteme anti-fraudă utilizate pentru protejarea şi furtul autoturismului sau a bunurilor aflate în interiorul său. Sistemul cuprinde sisteme senzoriale complexe (de tip web-cam) amplasate în interiorul/exteriorul maşinii, iar atunci când cineva doreşte să pătrundă în interiorul autovehiculului, este verificat dacă are acces să pătrundă sau nu; persoanele care au acces în acel autovehicul au codul irisului memorat într-o bază de date; dacă persoana are codul în baza de date i se permite să deschidă uşile sau să pornească motorul, altfel, se transmit frame-uri la Poliţie, firme de pază sau la proprietarul autovehiculului şi se intervine în timp util.

Un sistem biometric este, în esenţă, un sistem de recunoaştere a unor şabloane prin intermediul cărora este posibilă identificarea, aceasta fiind realizată prin stabilirea autenticităţii unor caracteristici biologice şi/sau comportamentale ale utilizatorului, a persoanei identificate. Din punct de vedere logic, un sistem biometric poate fi divizat în două unităţi funcţionale, module: un modul de înregistrare (enrollment) şi unul de identificare.

Modulul de înregistrare pregăteşte datele ce vor fi utilizate în sistem în vederea identificării persoanei, pe durata procesului scanându-se de către unul sau mai mulţi senzori anumite caracteristici biometrice, de exemplu amprentele digitale şi irisul, în vederea obţinerii digitale ale acestora. Ulterior, un program care este o componentă software cunoscută drept extractor de caracteristică, procesează reprezentările digitale achiziţionate de senzori pentru a genera forme compactizate ale acestora, denumite şabloane (template) care, dacă trăsăturile biometrice scanate sunt amprentele digitale şi irisul, pot conţine vârfuri de creste sau bifurcaţii (amprenta digitală) sau codul pentru iris. Şablonul fiecărui utilizator este memorat într-o bază de date, într-o reţea neuronală sau pe un smart card (o cartelă inteligentă din plastic ce conţine un microcip şi care poate stoca date cu caracter personal), caz în care cardul este înmânat utilizatorului. Pentru a fi identificat drept utilizator valid, deţinătorul cardului trebuie să aibă aceleaşi caracteristici cu cele memorate pe card.

Modulul de identificare realizează recunoaşterea persoanei. Pe durata identificării este scanată trăsătura bimetrică a persoanei ce trebuie identificată şi convertită într-o reprezentare digitală cu un format identic cu cel al şablonului folosit pentru memorarea în baza de date sau în reţeaua neuronală. Această reprezentare este, prin intermediul unuia sau a mai multor programe, cunoscută sub denumirea de comparator de caracteristici, comparat cu şablonul memorat într-o bază de date sau într-o reţea neuronală.

Un sistem de identificare va aprecia persoana drept corect identificată în momentul în care reprezentările trăsăturii scanate şi şablonul memorat sunt identice, altfel persoana va fi respinsă.

Un sistem de recunoaştere va aloca utilizatorului identitatea asociată şablonului din baza de date sau din reţeaua neuronală care a corespuns reprezentării trăsăturii scanate; dacă reprezentarea nu corespunde nici unuia dintre şabloane, persoana va fi rejectată.

(\*\*\*, 2018, )

## Dactiloscopie

Dactiloscopia (din greacă daktylos "deget" și skopia "examinare") este o metodă de identificare a persoanelor, folosită în antropologie, medicina legală și criminalistică care se bazează pe analizarea caracteristicilor reliefului papilar (amprentelor digitale), în urma cercetărilor antropologului englez Francis Galton care a demonstrat că desenele papilare digitale sunt strict individuale.

Relieful papilar se formează în perioada intrauterină a fătului și este alcătuit din creste și șanțuri care formează așa numitele desene papilare. Desenele papilare, specifice pielii corpului omenesc, aflate la nivelul degetelor, palmei și tălpii piciorului (plantă), cunoscute sub denumirea de dermatoglife, sunt formate din sistemul liniilor paralele ale crestelor papilare, separate între ele de șanțurile papilare. Sustenabilitatea folosirii metodei este dată de proprietățile desenelor reliefului papilar și anume de unicitatea și stabilitatea acestora. Unicitatea este dată de caracteristicile nivelului 2 și 3 care, prin plasamentul, forma și dimensiunea acestora, exclud repetabilitatea unui anumit tip papilar în ansamblul său. Fixitatea desenelor reliefului papilar face referire la permanența în timp (relieful papilar poate fi analizat după moartea omului, până la distrugerea acestuia prin putrefacție) a celor 3 nivele de caracteristici precum și la imobilitatea acestora.

Dactiloscopia ajută la identificarea persoanelor înregistrate în cazierul judiciar și la stabilirea antecedentelor penale ale acestora. Este un domeniu al criminalisticii alături de traseologie și balistică judiciară.

Utilizarea desenului papilar ca cea mai importantă metodă de identificare a persoanei se datorează următoarelor proprietăți:

* unicitatea desenului papilar, potrivit căreia amprentele se deosebesc între ele prin formă și detalii caracteristice, al căror număr și varietate fac imposibilă întâlnirea a două amprente identice (chiar și la aceeași persoană).
* fixitatea desenului papilar constă în menținerea tuturor detaliilor caracteristice ale desenului papilar de la formarea sa în a 13-a săptămână de viață intrauterină și până la moartea persoanei. Singura modificare, fără implicații în procesul identificării, o reprezintă creșterea în dimensiuni a amprentei, pe măsura dezvoltării corpului.
* inalterabilitatea desenului papilar, determinată de faptul că, în mod normal, un desen papilar nu poate fi modificat sau înlăturat.

În criminologie, expertiza dactiloscopică are următoarele etape:

* căutarea și descoperirea urmelor papilare la fața locului;
* examinarea și interpretarea la fața locului a urmelor papilare;
* relevarea și ridicarea urmelor papilare descoperite la fața locului;
* prelucrarea și examinarea urmelor papilare ridicate;
* obținerea amprentelor papilare model de comparație;
* identificarea persoanei care a creat urmele papilare prin efectuarea unor examinări dactiloscopice comparative.

Un specialist în dactiloscopie se numește dactiloscop.

Aprecierea valorii caracteristicilor în scopul identificării se bazează pe asocierea criteriilor calitative și cantitative, prioritate având cele calitative. Asemănările nu se numără, ci se „ponderează”. Identificarea depinde de natura, poziția și numărul asemănărilor. Absența deosebirilor nu dovedește prin ea însăși identitatea. În expertiza criminalistică nu s-a stabilit un număr de caracteristici necesare pentru individualizarea unui obiect după reflectarea sa și nici un număr minim de asemănări pentru stabilirea identității. Excepție face „regula celor 12 puncte” a lui Balthazard, aplicabilă în dactiloscopie (această regulă este uzitată în numeroase state europene și din America de Sud, precum și în Japonia. La polul superior al numărului de puncte coincidente necesare unui concluzii de identitate se află Italia – cu 16-17 puncte, iar a cel inferior Africa de Sud – 7 puncte, zona mediană – cu 8-12 puncte – se aplică în state precum Germania, Suedia, Olanda, Elveția.

(\*\*\*, 2018, )

## Conexiune Wi-Fi

Wi-Fi este numele comercial pentru tehnologiile construite pe baza standardelor de comunicație din familia IEEE 802.11 utilizate pentru realizarea de rețele locale de comunicație (LAN) fără fir (wireless, WLAN) la viteze echivalente cu cele ale rețelelor cu fir electric de tip Ethernet. Suportul pentru Wi-Fi este furnizat de diferite dispozitive hardware, și de aproape toate sistemele de operare moderne pentru calculatoarele personale (PC), rutere, telefoane mobile, console de jocuri și cele mai avansate televizoare.

Standardul IEEE 802.11 descrie protocoale de comunicație aflate la nivelul gazdă-rețea al Modelului TCP/IP, respectiv la nivelurile fizic și legătură de date ale Modelului OSI. Aceasta înseamnă că implementările IEEE 802.11 trebuie să primească pachete de la protocoalele de la nivelul rețea (IP) și să se ocupe cu transmiterea lor, evitând eventualele coliziuni cu alte stații care doresc să transmită.

802.11 face parte dintr-o familie de standarde pentru comunicațiile în rețele locale, elaborate de IEEE, și din care mai fac parte standarde pentru alte feluri de rețele, inclusiv standardul 802.3, pentru Ethernet. Cum Ethernet era din ce în ce mai popular la jumătatea anilor 1990, s-au depus eforturi ca noul standard să fie compatibil cu acesta, din punctul de vedere al transmiterii pachetelor.

Standardul a fost elaborat de IEEE în anii 1990, prima versiune a lui fiind definitivată în 1997. Acea versiune nu mai este folosită de implementatori, versiunile mai noi și îmbunătățite 802.11a/b/g fiind publicate între 1999 și 2001. Din 2004 se lucrează la o nouă versiune, intitulată 802.11n și care, deși nu a fost definitivată, este deja implementată de unii furnizori de echipamente.

Din punct de vedere al securității, IEEE și Wi-Fi Alliance recomandă utilizarea standardului de securitate 802.11i, respectiv a schemei WPA2. Alte tehnici simple de control al accesului la o rețea 802.11 sunt considerate nesigure, cum este și schema WEP, dependentă de un algoritm de criptare simetrică, RC4, nesigur.

Limitările standardului provin din mediul fără fir folosit, care face ca rețelele IEEE 802.11 să fie mai lente decât cele cablate, de exemplu Ethernet, dar și din folosirea benzii de frecvență de 2,4 GHz, împărțită în 12 canale care se suprapun parțial două câte două. Limitările date de consumul mare de energie, precum și de reglementările privind puterea electromagnetică emisă, nu permit arii de acoperire mai mari de câteva sute de metri, mobilitatea în cadrul acestor rețele fiind restrânsă. Cu toate acestea au apărut și unele tehnologii care permit legături fără fir bazate pe standardul 802.11 între două puncte fixe aflate la distanțe de ordinul sutelor de kilometri.

Spre deosebire de Ethernet, mediul de transmisie aduce probleme de securitate suplimentară. Dacă în Ethernet, accesul la cablu se putea restricționa prin ascunderea sau asigurarea zonelor prin care trece acesta, undele radio sunt mult mai dificil de controlat. Există mecanisme de bruiaj, care generează un zgomot electromagnetic ce acoperă frecvențele folosite de rețelele 802.11, dar acestea nu pot funcționa perfect, fără a afecta comunicațiile legitime sau fără a lăsa breșe prin care se poate obține acces în rețea. Cum la nivel fizic securitatea este dificil de asigurat, pentru obținerea unui nivel de securitate acceptabil este obligatorie criptarea datelor și controlul accesului la nivelele superioare celui fizic.

Rețelele Wi-Fi oferă rate de transfer mari pe distanțe mici, fiind una dintre cele mai rapide rețele fără fir, dar și cea care oferă cea mai mică mobilitate. Bluetooth este și ea o tehnologie fără fir gândită pentru acces pe distanțe și mai mici, de ordinul a câțiva metri, dar care oferă aceleași rate de transfer, avantajul ei constând într-o mai mare economie de energie. Pentru mărirea ariei de acoperire, IEEE a standardizat tehnologia WiMAX, pentru rețele metropolitane fără fir (IEEE 802.16). Întrucât echipamentele pentru WiMAX sunt însă costisitoare, Intel a dezvoltat o tehnologie de conexiune fără fir bazată pe standardele 802.11 cu dirijarea undelor, și care poate asigura legătura între două puncte aflate la distanțe de ordinul sutelor de kilometri, costul echipamentelor fiind cu două ordine de mărime mai mici decât cele ale punctelor de acces 802.16 (WiMAX).

(\*\*\*, 2018, )

## Programare orientată pe obiecte

Programarea orientată pe obiecte (POO, uneori și Programarea orientată obiect sau chiar denumită ca și în limba engleză, Object Oriented Programming (OOP)) este o paradigmă de programare.

Un obiect este unitatea de bază a programării orientate pe obiecte.

O clasă definește în ce constă un obiect al clasei și ce operații pot fi efectuate pe un astfel de obiect.

Principii de bază:

* Abstractizarea

Prin abstractizare se înțelege posibilitatea ca un program să ignore unele aspecte ale informației pe care o manipulează, adică posibilitatea de a se concentra asupra esențialului. Fiecare obiect în sistem are rolul unui “actor” abstract, care poate executa acțiuni, își poate modifica și comunica starea și poate comunica cu alte obiecte din sistem fără a dezvălui cum au fost implementate acele facilități.

* Încapsularea

Ascunderea de informații, asigură faptul că obiectele nu pot schimba starea internă a altor obiecte în mod direct (ci doar prin metode puse la dispoziție de obiectul respectiv); doar metodele proprii ale obiectului pot accesa starea acestuia. Fiecare tip de obiect expune o interfață pentru celelalte obiecte care specifică modul cum acele obiecte pot interacționa cu el.

* Polimorfismul

Abilitatea de a procesa obiectele în mod diferit, în funcție de tipul sau de clasa lor se numește polimorfism. Mai exact, este abilitatea de a redefini metode pentru clasele derivate.

* Moștenirea

Moștenirea organizează și facilitează polimorfismul și încapsularea, permițând definirea și crearea unor clase specializate plecând de la clase (generale) deja definite - acestea pot împărtăși (și extinde) comportamentul lor, fără a fi nevoie de a-l redefini. Aceasta se face de obicei prin gruparea obiectelor în clase și prin definirea de clase ca extinderi ale unor clase existente. Conceptul de moștenire permite construirea unor clase noi, care păstrează caracteristicile și comportarea, deci datele și funcțiile membru, de la una sau mai multe clase definite anterior, numite clase de bază, fiind posibilă redefinirea sau adăugarea unor date și funcții noi. O clasă moștenitoare a uneia sau mai multor clase de bază se numește clasă derivată.

(\*\*\*, 2016, )

## Principiile SOLID

Principiile de proiectare și programare orientată pe obiect s-au dezvoltat în timp, pe baza problemelor practice şi a experienței dobândite de către alți programatori. Aceste principii au fost formulate şi standardizate, în cele din urmă rezultând o serie destul de lungă de principii de design.

Dintre acestea, cele mai aplicate şi cunoscute sunt cele cinci care au fost rezumate sub acronimul SOLID (Single responsibility, Open-closed, Liskov substitution, Interface segregation şi Dependency inversion).

* Principiul responsabilității unice (Single responsibility principle)

Orice context (clasă, funcție, variabilă etc.) trebuie să aibă o unică responsabilitate, iar această responsibilitate trebuie să fie în întregime încapsulată în context. Toate serviciile sale trebuie să fie orientate pentru a servi această unică responsabilitate.

* Principiul deschis/închis (Open-closed principle)

Acest principiu implică ca entitățile software (clase, module, metode) să fie deschise pentru extensie dar închise pentru modificare Astfel, o entitate îşi poate extinde comportamentul fără modificari în codul sursă.

* Principiul de substituție Liskov (Liskov substitution principle)

Dacă S este un subtip al lui T, atunci obiectele de tip T pot fi substituite cu obiecte de tip S fără a afecta niciuna din proprietățile programului (corectitudine, realizarea execuției etc.).

* Principiul de segregare a interfețelor (Interface segregation principle)

Acest principiu afirmă că niciun client nu trebuie să fie forțat să depindă de metode pe care nu le utilizează. Interfețele trebuie separate în alte interfețe mai mici şi mai specifice.

* Principiul de inversare a dependențelor (Dependency Inversion principle)

Modulele de înalt nivel nu ar trebui să depindă de modulele de nivel scăzut. Ambele ar trebui să depindă de abstracții.

Cele cinci principii grupate sub eticheta SOLID pot fi aplicate in implementarea de software, în faza de design, sau pot fi utilizate pentru refactoring. Este o parte a strategiei globale de dezvoltare de programe solide şi extensibile.

(\*\*\*, 2018, )

## Modelul arhitectural MVC



Model-view-controller (MVC) (din engleză, aproximativ: model-vizualizare-controlor) este un model arhitectural utilizat în ingineria software. Succesul modelului se datorează izolării logicii de business față de considerentele interfeței cu utilizatorul, rezultând o aplicație unde aspectul vizual sau/și nivelele inferioare ale regulilor de business sunt mai ușor de modificat, fără a afecta alte nivele.

* Model

Această parte a controlatorului manipulează operațiunile logice și de utilizare de informație (trimisă dinainte de către rangul său superior) pentru a rezulta de o formă ușor de înțeles.

* Vizualizare

Acestui membru al familiei îi corespunde reprezentarea grafică, sau mai bine zis, exprimarea ultimei forme a datelor: interfața grafică ce interacționează cu utilizatorul final. Rolul său este de a evidenția informația obținută până ce ea ajunge la controlor.

* Controlor

Cu acest element putem controla accesul la aplicația noastră. Pot fi fișiere, scripturi (eng. scripts) sau programe, in general orice tip de informație permisă de interfață. În acest fel putem diversifica conținutul nostru de o formă dinamică și statică, în același timp.

Cu ajutorul Controlorului, modelului sau a parții de vizualizare putem manipula următoarele elemente: datele. Depinde de noi cum manipulăm și interpretăm aceste "date". Acum cunoaștem că unicele date ale unei adrese web statice sunt: obținerea unui fișier de pe disc(hard disk) sau din Internet, etc. și, interpretat (recunoscut/decodificat) sau nu, serverul răspunde.

Modelul, precum controlorul și vizualizarea (interfața grafică) manipulează toate datele ce se relaționeză cu el. Și numai partea de Vizualizare poate demonstra această informație. În acest fel am demonstrat ierarhia programului nostru: Controlor-Model-Vizualizare.

(\*\*\*, 2018, )

## Dezvoltarea aplicațiilor web

O aplicație web este un program care rulează într-o arhitectură client-server folosind tehnologiile deschise World Wide Web. Ele înlocuiesc modelele în care atât serverul cât și clientul rulează tehnologii proprietar, mentenanța aplicațiilor de pe partea de client fiind prea complexă, costisitoare și susceptibilă la erori. În schimb, omniprezența browserelor web și comoditatea de a le folosi drept client conduce la eliminarea acestei mari probleme. Astfel, au evoluat și pe partea de server serverele de aplicații, iar pe lângă limbajele de programare au apărut frameworkuri și tehnologii dedicate programării acestora.

Webul a apărut la început cu scopul de a distribui documente, inițial în cadrul său apărând protocolul HTTP și formatul HTML. Prima dezvoltare în direcția folosirii unui server web a fost apariția CGI⁠(en) la începutul anilor 1990. CGI permitea rularea unor scripturi pe server, care să genereze un răspuns dinamic în format HTML. Pasul următor a fost transformarea acestor documente HTML în interfețe dinamice cu utilizatorul, prin crearea de către Netscape a limbajului JavaScript, care este dedicat scriptării comportamentului browserului și introducerea de elemente dinamice; tot în acea perioadă, Sun a asociat limbajului Java capabilitatea de a rula mici aplicații scrise în acest limbaj și denumite appleți într-un plugin al browserului, iar Macromedia a inventat Flash, o tehnologie ce făcea posibil același lucru. În ultimii ani ai secolului al XX-lea, funcționalitatea limbajului JavaScript a fost extinsă semnificativ prin introducerea API-ului XmlHttpRequest, prin care se putea programa efectuarea și procesarea răspunsurilor la mici cereri HTTP dintr-o pagină web, fără a reîncărca întreaga pagină. Inițial, datele acestor mici cereri erau transferate de la client la server predominant în format XML, ceea ce a dus la denumirea de AJAX (Advanced JavaScript And XML), deși ulterior, pe măsură ce limbajele de programare server-side au introdus suportul, a început să fie preferat standardul JSON (obiecte JavaScript serializate).

Pe partea de server, CGI a fost urmat de apariția specificației Java Servlet, inclusă apoi în Java Enterprise Edition, ceea ce a făcut ca Java să devină limbaj de programare dominant în cadrul aplicațiilor web pe partea de server. Alte limbaje de programare dedicate programării server-side au fost ASP (dezvoltat de Microsoft) și PHP (proiect open-source), și apoi Ruby.

Anii 2010 au adus o mult-așteptată extindere a standardului HTML, cu HTML5, care impune browserelor implementarea internă a multor funcționalități care până atunci fuseseră apanajul exclusiv al pluginurilor dezvoltate de terți, iar dezvoltarea explozivă a smartphone-urilor a dus mai întâi la o lipsă de browsere web cu funcționalități similare celor pentru calculatoare și deci la dezvoltarea de aplicații client dedicate fiecărei aplicații web.

(\*\*\*, 2018, )

## Controlul versiunilor

Controlul versiunilor (din engleză: version control sau revision control) este un domeniu software care se ocupă cu gestionarea mai multor versiuni (numite și revizii) ale unor fișiere. Este aplicată cu predilecție în programare, cu scopul de a păstra versiuni succesive ale codului sursă al unui program de calculator. O soluție ar fi arhivarea separată și completă a fiecărei versiuni a programului într-o bază de date (pe un purtător de date extern), dar această metodă ar necesita în general prea mult spațiu de memorie. În locul ei se utilizează tehnici speciale, care reduc memoria totală necesară și care facilitează reconstrucția „în zbor”, la cerere, a oricărei versiuni din istoria programului.

Prima generație de unelte pentru controlul versiunilor foloseau/versionau câte un singur fișier și nu aveau o corespundere între diferite fișiere din repository. Acestea nu aveau suport pentru rețea.

Exemple de astfel de unelte: Source Code Control System (SCCS), Revision Control System (RCS).

A doua generație de unelte pentru controlul versiunilor folosesc/versionează mai multe fișiere si aveau o corespundere directă între ele. Acestea erau centralizate.

Exemple de astfel de unelte: Concurrent Versions System (CVS), Subversion (SVN), TFS, Perforce, SVK, VSS.

A treia generație de unelte pentru controlul versiunilor folosesc/versionează mai multe fișiere si aveau o corespundere directă între ele dar sunt descentralizate.

Exemple de astfel de unelte: git, BitKeeper (BK), Bazaar.

Terminologie:

* repository - „depozitul“ în care sunt păstrate fișierele curente și versiunile anterioare. Deseori acest depozit este o bază de date găzduită pe un server.
* working copy (copie de lucru) - copie a fișierelor din repository pe calculatorul de lucru al unui dezvoltator (de unde și numele). Acestea sînt fișierele pe care lucrează un dezvoltator în mod obișnuit.
* check-out - operația de creare a unei copii de lucru luate din repository
* commit sau check-in - operația de introducere în repository a schimbărilor din copia de lucru
* update (actualizare) - introducerea în copia de lucru a schimbărilor făcute de alte persoane (colegi la același proiect) la repository
* branch (ramificare) - bifurcarea unui set de fișiere în două căi de dezvoltare distincte
* merge (integrare) - unirea a două versiuni diferite ale unui aceluiași fișier într-o singură versiune
* tag - o „etichetă“ aplicată fișierelor din repository la un anumit moment important din "viața" programului, de exemplu la lansarea unui produs

(\*\*\*, 2018, )

## Imprimare 3D

Imprimarea 3D se referă la diferite procese utilizate pentru a sintetiza un obiect tridimensional. În imprimarea 3D, straturi succesive de material sunt formate sub control computerizat pentru a crea un obiect. Aceste obiecte pot fi de aproape orice formă sau geometrie și sunt fabricate dintr-un model 3D sau alte surse de date electronice. O imprimantă 3D este un tip de robot industrial.

Prin printare 3D se pot realiza mecanisme pe deplin funcționale, dintr-un singur proces tehnologic, fără a mai fi necesare alte procedee de asamblare post-producție.

Pentru cazul în care vor exista spații între părți în mișcare este folosit un al doilea material denumit material suport. Odată ce piesa dorită este finalizată aceasta este scoasă din mașină, iar materialul suport este eliminat mecanic sau, după caz, dizolvat în cuva cu ultrasunete, lăsând un mecanism de lucru perfect.

Avantajul major față de metodele tradiționale de fabricație:

* reducerea timpului de realizare
* rapiditate
* cost raportat doar la volum
* realizarea formelor imposibile pentru tehnologia tradițională
* înaltă acuratețe

(\*\*\*, 2016, )

(\*\*\*, 2016, )

# Soluţia propusă şi metodologia de proiectare și dezvoltare

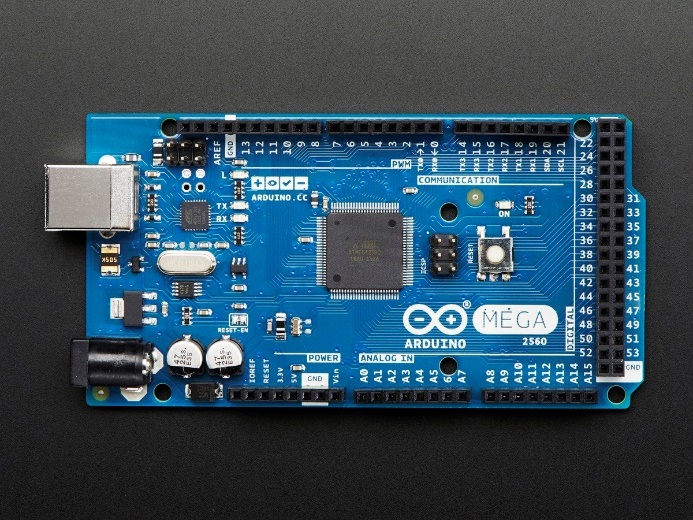
## Satisfacerea cerințelor de funcționare

### Dispozitivul de pontare

### Aplicația web

## Componente

### Arduino Mega 2560



Acest micontroler este proiectat pentru proiecte mai complexe.

Arduino Mega 2560 este un microcontroler bazat pe ATmega2560, dispune de 54 de intrări / ieșiri digitale dintre care 15 pot fi utilizate ca ieșiri PWM, 16 intrări analogice, 4 porturi seriale hardware UART, un oscilator de cristal de 16 MHz, o conexiune USB, o mufă de alimentare, un circuit de programare serială ICSP și un buton de resetare.

Placa Mega 2560 este compatibilă cu majoritatea shieldurilor proiectate pentru Uno și fostele plăci Duemilanove sau Diecimila.

Conectarea acestuia la un calculator se face prin intermediul unui cablu de tip USB.

Cu ajutorul placii de dezvoltare Arduino Mega 2560 se pot concepe diverse proiecte începând de la aprinderea unor leduri până la construirea unor adevarați roboți.

Programarea microcontrolerului se poate face cu Arduino Software (IDE).

Ca masură de precauție, Arduino Mega protejează porturile USB ale calculatorului de scurtcircuitare sau supracurent cu o siguranță resetabilă. Siguranța furnizează o protecție suplimentară deoarece majoritatea calculatoarelor își asigură propria lor protecție internă. Siguranța va întrerupe automat conexiunea până când suprasarcina este înlăturată din situația în care asupra portului USB se aplică mai mult de 500 mA.

Alimentarea microcontrolerului se face fie prin conectare USB, fie printr-o sursă de alimentare externă, aceasta fiind selectată automat. Sursa de alimentare externă poate fi un adaptor AC-DC sau o baterie. Conectarea adaptorului se face prin mufa de alimentare iar conectarea bateriei se face cu ajutorul pinilor GND si Vin.

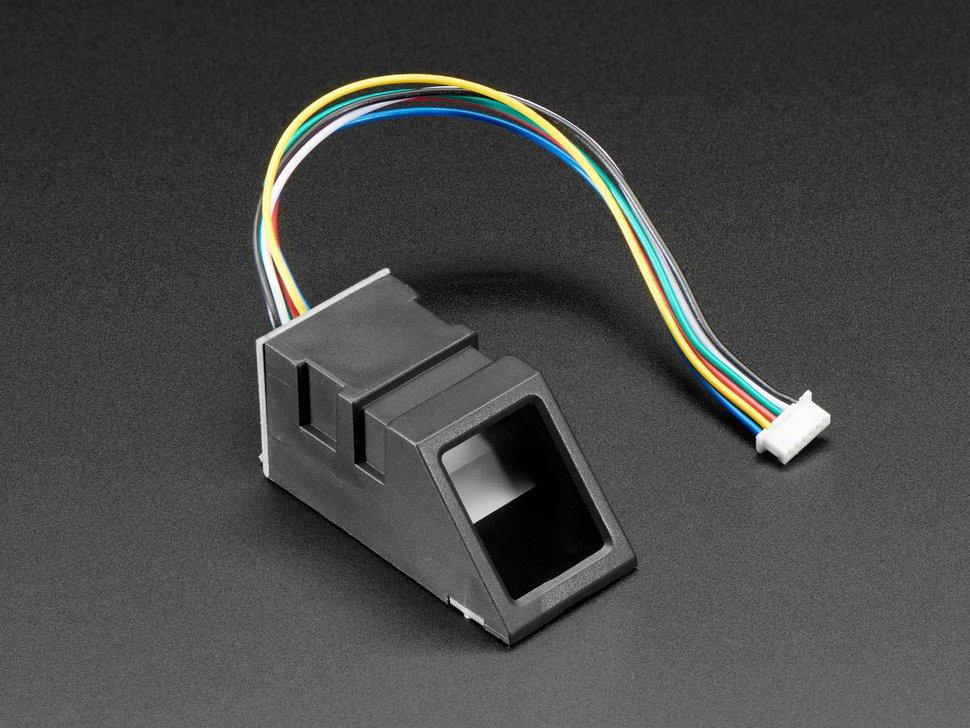
Placa poate funcționa cu o sursă externă de 6 până la 20 de volți. Dacă este alimentată cu mai puțin de 7V, pinul 5V va furniza mai puțin de 5 volți, iar placa poate deveni instabilă. Dacă se utilizează mai mult de 12V, regulatorul de tensiune se poate supraîncălzi și poate deteriora placa. Intervalul recomandat este de la 7 la 12 volți.

Specificații tehnice:

|  |  |
| --- | --- |
| Microcontroler | Atmega2560 |
| Tensiune de lucru | 5 V |
| Tensiune de intrare (recomandată) | 7-12 V |
| Tensiune de intrare (limita) | 6-20 V |
| Pini digitali de intrare/ieșire | 54 din care 15 pot fi utilizați ca ieșiri PWM |
| Pini digitali PWM de intrare/ieșire | 15 |
| Pini analogici de intrare | 16 |
| Curent continuu pentru pinii de intrare/ieșire | 20 mA |
| Curent continuu pentru pinii de 3.3V | 50 mA |
| Memorie flash | 256 KB |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Viteza ceasului | 16 MHz |
| LED\_BUILTIN | 13 |
| Lungime | 101,52 mm |
| Lățime | 53.3 mm |
| Greutate | 37 g |

(\*\*\*, 2018, )

### Senzor de amprentă



Acest senzor poate oferi securitate unui proiect prin intermediul elementelor biometrice. Senzorul optic de amprentă digitală face posibilă detectarea și verificarea amprentelor digiale. Modulele de acest tip sunt în mod obișnuit folosite în seifuri.

Cipul performant DSP (digital signal processor) integrat realizează redarea imaginilor, procesarea, calcularea, găsirea de elemente și căutarea lor.

Prin conectarea modulului la orice microcontroller sau sistem cu comunicare serială TTL (transistor-transistor logic) se trimit pachete de date pentru a face capturi de imagini, pentru a detecta amprentele, pentru procesare și pentru căutare.

Modulul permite înregistrarea directă a unor noi amprente cu posibilitatea de stocare de până la 162 de amprente în memoria flash integrată.

Un LED roșu se aprinde pe cadranul senzorului cand acesta capturează o imagine.

Senzorul de amprentă nu numai că este ușor de utilizat, dar vine și cu un software pentru Windows, care face ca testarea modulului să fie simplă - permite înregistrarea și vizualizarea imaginii amprentei pe ecranul calculatorului.

Modulul dspune atât de o bibliotecă Arduino cât și o bibliotecă CircuitPython, tutoriale detaliate, instrucțiuni de cablare, conectare și utilizare, astfel încât să fie folosit chiar și de începători.

Modulul efectuează o serie de funcții cum ar fi înregistrarea amprentelor digitale, procesarea imaginilor, identificarea amprentelor digitale, stocarea șablonului și căutarea.

Procesarea amprentelor digitale include două părți: înscrierea amprentelor și identificarea amprentelor (potrivirea poate fi 1 : 1 sau 1 : N). La înscriere, utilizatorul trebuie să introducă degetul de două ori. Sistemul va procesa cele două imagini ale degetului, va genera un șablon al degetului pe baza procesării rezultatelor și va stoca șablonul. Atunci când se potrivește, utilizatorul introduce degetul prin senzorul optic și sistemul generează un șablon al degetului și îl compară cu șabloanele bibliotecii degetului. Pentru potrivirea de 1 : 1, sistemul va compara degetul cu șablonul specifc desemnat în modul. Pentru potrivirea de 1 : N sau pentru căutare, sistemul va căuta întreaga bibliotecă de degete pentru degetul de potrivire. În ambele cazuri, sistemul va returna rezultatul care se potrivește, succesul sau eșecul.

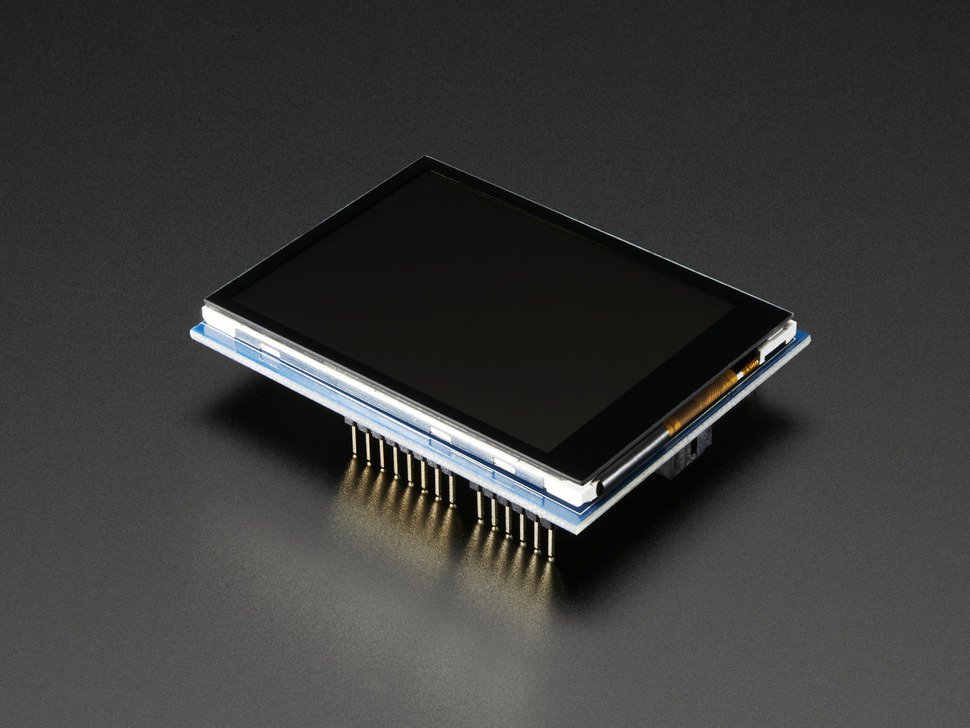
Acesta este de departe cel mai bun senzor de amprente disponibil în momentul de față.

Specificații tehnice:

|  |  |
| --- | --- |
| Tensiune de alimentare | 3.6 – 6 V DC |
| Curent de funcționare normal | 100 mA |
| Curent de funcționare de vârf | 150 mA |
| Rata baud | (9600\*N) bps, N = 1 ~ 12 (default N = 6) |
| Timp de capturare imagine | < 1 s |
| Capacitate de stocare | 120 / 375 / 880 |
| FAR | < 0.001 % |
| FRR | < 0.1 % |
| Timp mediu de căutare | < 1 s (1 : 880) |
| Temperatură de funcționare | -10 – +40 °C |
| Umiditate de funcționare | 40 – 85 %RH |
| Temperatură de depozitare | -40 – +85 °C |
| Umiditate de depozitare | < 85 %RH |
| Dimensiune modul | 42 x 25 x 8.5 mm |
| Dimensiune instalare | 31.5 x 19 mm |
| Dimensiune senzor | 56 x 20 x 21.5 mm |
| Dimensiune cadran | 14 x 18 mm |
| Interfață | UART (TTL logical level) / USB 1.1 |
| Mod potrivire | 1 : 1 și 1 : N |
| Dimensiune fișier caracter | 256 bytes |
| Dimensiune șablon | 512 bytes |
| Nivel securitate | 5 (1, 2, 3, 4, 5 -cel mai ridicat nivel) |

(\*\*\*, 2018, )

### 2.8" TFT Touchscreen Shield



Ecranul tactil este un shield dedicat pentru Arduino cu un ecran frumos mare de afișare cu ecran tactil, cu un slot de card microSD integrat și un ecran tactil capacitiv.

Acest afișaj TFT are o luminozitate mare, diagonală de 2.8”, o paletă de 262000 de nuanțe diferite de culori, 240 x 320 pixeli, cu un control individual al pixelilor. Bonus, acest ecran are un touchscreen capacitiv atașat, astfel încât să detecteze apăsarea degetelor oriunde pe ecran.

Acest shield este versiunea capacitivă a ecranului tactil rezistiv. Acest ecran tactil nu necesită apăsarea pe ecran cu un stilou și are o acoperire din sticlă lucioasă. Este un afișaj ce suportă o singură atingere.

Acest touchscreen utilizează SPI pentru afișare și dispune de slot pentru card microSD și este compatibil cu Arduino UNO, Mega și Leonardo. Controlerul capacitiv cu touchscreen utilizează I2C, dar permite partajarea magistralei I2C și cu alte dispozitive I2C.

Ecranul este complet asamblat, testat și gata de utilizare. Nu necesită cablare sau lipire, pur și simplu se conectează iar prin intermediul bibliotecii este ușor de pus in funcționare.

Cel mai bine funcționează cu Arduino UNO, Duemilanove, Diecimila. Pentru Arduino Leonardo sau Mega se lipesc trei jumperi pentru a-l folosi la viteză maximă.

Acest shield de afișare are un controler integrat cu propria memorie RAM, astfel încât aproape nici o operație nu se face de către microcontroler. Acest ecran necesită mai puțini pini astfel încât să permită conectarea mai multor senzori, butoane și LED-uri: 5 pini SPI pentru afișaj, 2 pini I2C partajați pentru controlerul touchscreen și un alt pin pentru cardul microSD dacă se dorește citirea imaginilor de pe aceasta.

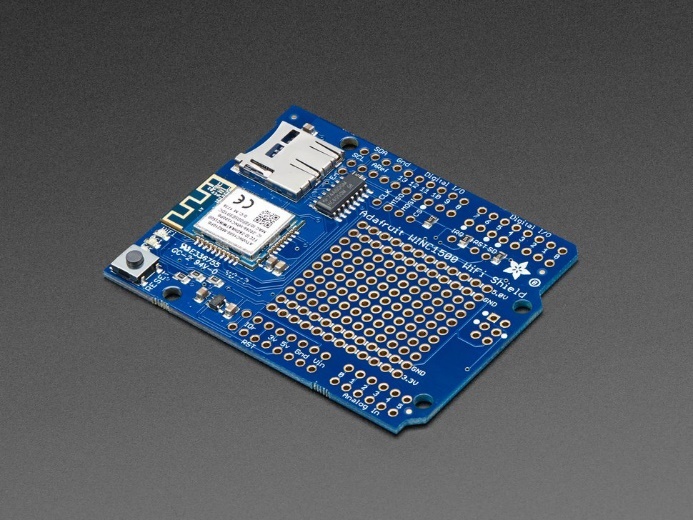
Ecranul dispune de o bibliotecă grafică completă open source care poate desena pixeli, linii, dreptunghiuri, cercuri și text și de asemenea, o bibliotecă pentru ecranul tactil care detectează locația x și y împreuna cu codul de exemplelor pentru a demonstra totul. Codul este scris pentru Arduino, dar poate fi ușor adaptat la microcontrolerul preferat.

Specificații tehnice:

|  |  |
| --- | --- |
| Tip LCD | 2.8” TFT cu CTP |
| Tip CTP | G + G un singur punct și gesturi |
| Aranjarea punctelor | 240 x 320 puncte RGB |
| Vector filtru color | Dungă verticală RGB |
| Mod afișaj | TN / Transmission / Normally White |
| Direcție vizualizare | Ora 6 |
| TFT Driver IC | ILI9341 |
| CTP Driver IC | FT6206 |
| Dimensiune modul | 50 x 69.2 x 4 mm |
| Zonă activă | 43.2 x 57.6 mm |
| Punct | 0.18 x 0.18 mm |
| Temperatură de funcționare | -20 – +70 °C |
| Temperatură de depozitare | -30 – +80 °C |
| Umiditate de depozitare | 20 – 90 %RH |
| Lumină de fundal | 4 leduri albe în paralel |
| Tensiune logică de alimentare  IOVCC | -0.3 – 4.6 V  Min = 1.65 V, Typ = 1.8 / 2.8 V, Max = 3.3 V |
| Tensiune analogică de alimentare  VCI | -0.3 – 4.6 V  Min = 2.5 V, Typ = 2.8 V, Max = 3.3 V |
| Tensiune de intrare | -0.3 – (IOVCC + 0.3) V |
| Tensiune ridicată de intrare | Min = 0.7IOVCC V, Max = IOVCC V |
| Tensiune scăzută de intrare | Min = GND V, Max = 0.3IOVCC V |
| Tensiune ridicată de ieșire | Min = 0.8IOVCC V, Max = IOVCC V |
| Tensiune scăzută de ieșire | Min = GND V, Max = 0.2IOVCC V |
| Pierderi de curent I/O | -0.1 – 0.1 uA |

(\*\*\*, 2018, )

### WiFi Shield



Microcontrolerul Arduino se poate conecta la internet prin intermediul acestui shield WiFi care dispune de un modul ATWINC1500 certificat de FCC de la Atmel. Acest modul Wi-Fi cu capacitate de 802.11bgn este cel mai bun pentru a conecta dispozitivele la internet datorită suportului SSL și performanței solide.

Adaptorul ATWINC1500 WiFi Shield utilizează SPI pentru a comunica plus GPIO pentru control, deci cu aproximativ 6 fire se realizează conectarea. În prezent, biblioteca furnizată de Atmel funcționează cel mai bine cu plăci bazate pe SAMD21, cum ar fi Arduino Zero, Metro M0 Express sau Arduino Mega. Funcționează bine și pe Uno, dar utilizează aproape toată memoria, astfel încât poate fi o provocare pentru a construi proiecte complexe. Acesta nu va funcționa pe alte plăcuțe Arduino.

Scanarea și conectarea la rețele este foarte rapidă și se realizează în câteva secunde.

Acest modul funcționează cu rețelele 802.11 b, g sau n și suportă criptarea WEP, WPA și WPA2. Se poate utiliza în modul Soft AP pentru a crea o rețea ad-hoc. Pentru conexiunile client securizate, există suport TLS 1.2.

Această versiune a shieldului vine cu o antenă integrată în PCB.

Aceste module WiFi de protocol SPI sunt în prezent cea mai bună modalitate de a adăuga WiFi unui microcontroler Arduino.

Există, de asemenea, 3 LED-uri care se pot controla peste interfața SPI, parte a codului bibliotecii, sau pot fi controlate de biblioteca Arduino. Acestea se vor aprinde atunci când sunt conectate la un SSID sau transmit date.

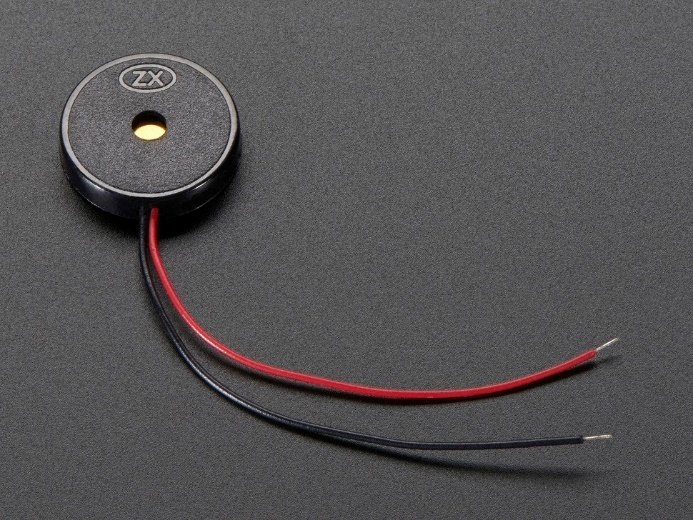
De asemenea, shieldul are integrat un card micro SD, care se poate utiliza pentru a stoca datele obținute de pe Internet.

Specificații tehnice:

|  |  |
| --- | --- |
| Putere de ieșire | 17 dBm |
| Interfața gazdă | SPI |
| Pini | 28 |
| Transceiver RF | da |
| Temperatură de funcționare | -40 – +80 °C |
| Interval frecvență | 2.412 – 2.472 MHz |
| Sensibilitate intrare Rx | -95 dB |
| Consum curent TX | 172 mA |
| Consum curent RX | 70 mA |
| Tensiune de operare | 3 – 4.2 V |
| SPI | 1 |
| UART | 1 |
| Parteneri Cloud | da |
| Antenă | PCB, U.FL |
| Bandă b/g/n | 2.4 GHz |
| Protocoale de securitate acceptate | WPA/WPA2 Personal, TLS, SSL |
| Servicii de rețea | DHCP, DNS, TCP/IP (IPv4), UDP, HTTP, HTTPS |
| Dimensiuni | 68 x 53 x 4.7 mm  2.7 x 2.1 x 0.2 ” |
| Greutate | 13.7 g / 0.5 oz |

(\*\*\*, 2018, )

### Piezo



Acest element piezo mic cu diametrul de 14 mm este bine închis, astfel încât permite să fie atașa cu ușurință. Elementele Piezo transformă vibrația la tensiune sau tensiunea la vibrații. Asta înseamna că pot funcționa ca un buzzer pentru a face bipuri, sunete si alerte si de asemenea ca senzor, pentru a detecta mișcari rapide cum ar fi loviturile. Poate servi drept senzor de avarie.

Este evaluat pentru o utilizare de până la 9Vpp, dar se poate utiliza, de asemenea, cu semnale treaptă de 3V și emite un volum ridicat.

Pentru a reda muzică cu un Arduino, există tutorialul Tone iar pentru detectare, tutorialul Knock. Componenta are atașate fire subțiri, introduse într-un panou lipit fără sudură, care s-ar putea să fie prea subțiri pentru a se conecta direct la pinii de la Arduino.

Specificații tehnice:

|  |  |
| --- | --- |
| Frecvența de rezonanță | 8 kHz |
| Max curent nominal | 3 mA |
| Tensiune nominală | 9 V |
| Temperatură de funcționare | -20 – +70 °C |
| Tensiune maximă de intrare | 40 V |
| Diametru extern | 14 mm / 0.55 “ |
| Centrul circumferință | 5.5 mm / 0.2 “ |
| Grosime | 2.5 mm / 0.1 “ |
| Greutate | 0.5 g |

(\*\*\*, 2018, )

### Buton On/Off Switch



Aceste butoane din metal cromate sunt rezistente și arată bine în timp. Se montează pur și simplu într-un orificiu de 16 mm în orice material cu grosimea de până la 1/2 “. Există chiar și o garnitură de cauciuc pentru a proteja de umiditatea mediului. Pe partea din față a butonului este un cerc plat din metal, înconjurat de un inel albastru din plastic. În partea din spate există 3 contacte pentru buton (comun, normal deschis și normal închis) și 2 pentru inelul LED albastru (+ și -).

Acest buton este un buton de pornire / oprire iar când apăsați contactele se închid iar butonul rămâne apăsat. Când îl apăsați a doua oară, butonul se deschide și contactele se deschid.

Comutatorul și LED-ul sunt separate, astfel încât se poate conecta la apăsare sau invers.

Specificații tehnice:

|  |  |
| --- | --- |
| Dimensiune orificiu montare | 16 mm |
| Valori nominale | 3 A / 250 VDC |
| Contacte | 1 NO, 1 NC |
| Material | Alamă cromată |
| Rezistența la contact | < 50 mOhm |
| Resiztența izolării | > 1000 mOhm |
| Temperatură de funcționare | -20 – +55 °C |
| Viața mecanică | > 500,000 |
| Viața electrică | > 50,000 |
| Grosimea panoului | 1–13 mm |
| Presiunea de lucru | 1.5 – 2.5 N |
| Ciclu de operare | 2 mm |
| Tensiunea nominală a lămpii | 6 V |
| Greutate | 13.9 g |

(\*\*\*, 2018, )

### Diverse

* Elemente de fixare: șuruburi
* Pini
* Conductori și conectori
* Suport baterie 9V
* Carcasă imprimată 3D

## Tehnologii folosite

### Programe software

#### Arduino IDE

Mediul integrat de dezvoltare Arduino conține un editor de text pentru scrierea de cod, o zonă de mesaj, o consolă de text, o bară de instrumente cu butoane si o serie de meniuri. Se conectează la hardwareul Arduino și Genuino pentru a încărca programe și de a comunica cu acestea.

Programele scrise folosind Arduino Software (IDE) sunt numite schițe. Aceste schițe sunt scrise în editorul de text și sunt salvate cu extensia de fișier .ino. Conceptul de – sketchbook (carte de schițe) este folosit drept loc standard pentru salvarea programelor. Arduino Software permite gestionarea mai multor schițe în paralel, fiecare deschisă în propria fereastră. Acestea pot fi fișiere Arduino (extensia .ino), fișiere C (extensia .c), C ++ fișiere (extensia .cpp) sau fișiere header (extensia .h).

Zona de mesaje oferă un feedback în timpul salvării și exportării și de asemenea afișează erori.

(\*\*\*, 2016, )

#### Visual Studio

Visual Studio include un set complet de instrumente de dezvoltare pentru generarea de aplicații ASP.NET, Servicii Web XML, aplicații desktop și aplicații mobile. Visual Basic, Visual C++, Visual C# și Visual J# toate folosesc același mediu de dezvoltare integrat (IDE) care le permite partajarea instrumentelor și facilitează crearea de soluții folosind mai multe limbaje de programare. Aceste limbaje permit să beneficieze de caracteristicile .NET Framework care oferă acces la tehnologii cheie care simplifica dezvoltarea de aplicații web ASP și XML Web Services cu Visual Web Developer.

(\*\*\*, 2018, )

#### Visual Studio Code

Visual Studio Code este un editor cross platform pentru Windows, MacOS și Linux. Cu ajutorul său orice dezvoltator va putea să scrie cod și să îl editeze și chiar să facă debugging direct din sistemul de operare pe care îl folosește cel mai mult.

Visual Studio Code oferă dezvoltatorilor suportul complet pentru o gamă variată de limbaje de programare și toate instrumentele de editare pe care un dezvoltator le așteaptă de la o aplicație modernă: evidențierea sintaxei, shortcut-uri personalizabile etc.

Microsoft a preferat să deschidă sursele numai pentru editorul său de cod multiplatformă web și cloud (IDE) și nu și pentru întregul pachet Visual Studio IDE.

(\*\*\*, 2018, )

(\*\*\*, 2018, )

#### SQL Server Management Studio

SQL Server Management Studio (SSMS) este o aplicație software lansată pentru prima dată cu Microsoft SQL Server 2005 care este utilizată pentru configurarea, gestionarea și administrarea tuturor componentelor din Microsoft SQL Server. Instrumentul include atât editori de script-uri, cât și instrumente grafice care funcționează cu obiecte și caracteristici ale serverului.

O caracteristică centrală a SSMS este Object Explorer, care permite utilizatorului să navigheze, să selecteze și să acționeze asupra oricăror obiecte din cadrul serverului. De asemenea, a fost livrată o ediție Express separată, care ar putea fi descărcată liber. Versiunile recente ale SSMS sunt pe deplin capabile să se conecteze și să gestioneze orice instanță SQL Server Express. Microsoft a incorporat, de asemenea, compatibilitate reversibilă pentru versiunile mai vechi ale SQL Server, permițând astfel ca o versiune mai nouă a SSMS să se conecteze la versiunile mai vechi ale instanțelor SQL Server.

Pornind de la versiunea 11, aplicația se baza pe shell-ul Visual Studio 2010, folosind WPF pentru interfața cu utilizatorul.

În iunie 2015, Microsoft și-a anunțat intenția de a lansa versiuni viitoare ale SSMS independent de versiunile de baze de date SQL Server.

(\*\*\*, 2018, )

#### Microsoft SQL Server

Microsoft SQL Server este un sistem de gestionare de baze de date relaționale (RDBMS) produs de compania americană Microsoft Corp. Limbajul de interogare este SQL iar extensia procedurală este T-SQL.

Suportă versiunea companiei Microsoft de SQL (Structured Query Language - limbaj structurat de interogări), cel mai răspândit limbaj de interogare pentru bazele de date. Este considerat un sistem de gestionare a bazelor de date pentru întreprinderi - adică se poate aplica bazelor de date de dimensiuni foarte mari - însă la ultimele versiuni au venit cu mai multe tipuri de licențiere, de la cele free la unele avansate, astfel că în prezent putem spune că SQL Server se poate folosi pentru baze de date de orice dimensiune.

Microsoft SQL Server folosește o variantă de SQL numită T-SQL, sau Transact-SQL, o implementare de SQL-92 (standardul ISO pentru SQL) cu unele extensii. T-SQL în principal adaugă sintaxă adițională pentru procedurile stocate și pentru tranzacții. Standardele SQL necesită ACID; acesta este un acronim pentru cele 4 condiții ce trebuie îndeplinite de orice tranzacție: atomicitate, consistență, izolare, durabilitate. MS SQL Server suportă ODBC (Open Database Connectivity).

Stocarea datelor se face pe modelul relațional, sub forma unei colecții de tabele cu rânduri și coloane. Fiecare coloană este de un anumit tip: SQL Server suportă atât tipurile primare, precum întreg, zecimal, caracter, dată calendaristică, dar și tipuri mai complexe precum text, date binare, date geometrice, date spațiale, XML, etc.

(\*\*\*, 2018, )

#### Microsoft IIS

Internet Information Services (IIS) - denumit anterior Internet Information Server - reprezintă un set de servicii de Internet destinate serverelor care folosesc sistemul de operare Microsoft Windows. Acesta este al doilea server web ca popularitate, fiind devansat doar de liderul acestei ramuri industriale, Apache HTTP Server. În prezent serverele includ protocoale de rețea precum FTP, FTPS, SMTP, NNTP și HTTP/HTTPS.

(\*\*\*, 2018, )

#### Tinkercad

Tinkercad este o aplicatie online,în care se poate proiecta 3D pentru imprimare 3D.

Tinkercad este folosit de designeri, pasionați, profesori și chiar copii, pentru a face jucării, prototipuri, obiecte de decor, modele Minecraft, bijuterii și lista este cu adevărat fără sfârșit.

Mod de funcționare:

* Plasare – Formele geometrice sunt blocuri de construcție de bază, avănd proprietatea de a adăuga sau elimina material.
* Ajustare – Schimbarea poziției, rotirea și redimensionarea in cele 3 planuri x,y și z sunt acțiunile ce se pot lua pe o formă. Pentru precizie se poate utiliza rigla.
* Combinare – Pentru a crea modele cât mai diversificate se folosește gruparea unui set de forme.

(\*\*\*, 2016, )

### Limbaje de programare

#### C

C este un limbaj de programare standardizat, implementat pe majoritatea platformelor de calcul din zilele noastre.

Apreciat pentru eficiența codului obiect generat de compilatoarele C, și pentru portabilitatea sa, este cel mai popular limbaj de programare pentru scrierea de software de sistem.

Sintaxa limbajului C a stat la baza multor limbaje create ulterior și încă populare azi precum: C++, Java, JavaScript, C#, D.

C este un limbaj de programare relativ minimalist ce operează în strânsă legătură cu hardwareul, fiind cel mai apropiat de limbajul de asamblare față de majoritatea celorlalte limbaje de programare.

Limbajul C aparține clasei limbajelor de nivel scăzut sau de nivel mediu, aceasta indicând strânsa legătură între interoperabilitate și echipamentul hardware.

C a fost creat având drept scop important de a face ca programele mari să poată fi scrise mai ușor și cu mai puține erori.

Caracteristicile importante constă în: funcționalități importante, paradigma programării procedurale, set simplu de tipuri de date, limbaj preprocesor, acces la nivel scăzut a memoriei calculatorului, folosirea parametrilor, forme rudimentare de închidere și polimorfism, declararea variabilelor și structuri de date sau tipuri de date agregate.

Bineînteles, pe lângă plusuri, C vine cu minusuri precum lipsa: unui sistem automat de colectare a reziduurilor, claselor și obiectelor, sistemului avansat de tipuri de date, programării generice și a procesării listelor.

(\*\*\*, 2016, )

#### HTML

HyperText Markup Language (HTML) este un limbaj de marcare utilizat pentru crearea paginilor web ce pot fi afișate într-un browser (sau navigator). Scopul HTML este mai degrabă prezentarea informațiilor precum paragrafe, fonturi, tabele ș.a.m.d. decât descrierea semanticii documentului.

Specificațiile HTML sunt dictate de World Wide Web Consortium (W3C).

HTML este o formă de marcare orientată către prezentarea documentelor text pe o singura pagină, utilizând un software de redare specializat, numit agent utilizator HTML, cel mai bun exemplu de astfel de software fiind browserul web. HTML furnizează mijloacele prin care conținutul unui document poate fi adnotat cu diverse tipuri de metadate și indicații de redare. Indicațiile de redare pot varia de la decorațiuni minore ale textului, cum ar fi specificarea faptului că un anumit cuvânt trebuie subliniat sau că o imagine trebuie introdusă, până la scripturi sofisticate, hărți de imagini și formulare. Metadatele pot include informații despre titlul și autorul documentului, informații structurale despre cum este împărțit documentul în diferite segmente, paragrafe, liste, titluri etc. și informații cruciale care permit ca documentul să poată fi legat de alte documente pentru a forma astfel hiperlink-uri (sau web-ul).

HTML este un format text proiectat pentru a putea fi citit și editat de oameni utilizând un editor de text simplu. Totuși scrierea și modificarea paginilor în acest fel solicită cunoștințe solide de HTML și este consumatoare de timp. Editoarele grafice (de tip WYSIWYG) cum ar fi Macromedia Dreamweaver, Adobe GoLive sau Microsoft FrontPage permit ca paginile web sa fie tratate asemănător cu documetele Word, dar cu observația că aceste programe generează un cod HTML care este de multe ori de proastă calitate.

HTML se poate genera direct utilizând tehnologii de codare din partea serverului cum ar fi PHP, JSP sau ASP. Multe aplicații ca sistemele de gestionare a conținutului, wiki-uri și forumuri web generează pagini HTML.

HTML este de asemenea utilizat în e-mail. Majoritatea aplicațiilor de e-mail folosesc un editor HTML încorporat pentru compunerea e-mail-urilor și un motor de prezentare a e-mail-urilor de acest tip. Folosirea e-mail-urilor HTML este un subiect controversat și multe liste de mail le blochează intenționat.

Paginile HTML sunt formate din etichete sau tag-uri și au extensia „.html” sau „.htm”. În marea lor majoritate aceste etichete sunt pereche, una de deschidere <eticheta> și alta de închidere </eticheta>, mai există și cazuri în care nu se închid, atunci se folosește <eticheta />. Navigatorul web interpretează aceste etichete afișând rezultatul pe ecran. HTML-ul este un limbaj care nu face deosebire între litere majuscule și minuscule.

Pagina principala a unui domeniu este fisierul „index.html” respectiv „index.htm”. Această pagină este setată a fi afișată automat la vizitarea unui domeniu.

De exemplu la vizitarea domeniului www.nume.ro este afișată pagina www.nume.ro/index.html.

Unele etichete permit utilizarea de atribute care pot avea anumite valori: <eticheta atribut="valoare"> ... </eticheta>

Componența unui document HTML este:

* versiunea HTML a documentului
* zona head cu etichetele <head> </head>
* zona body cu etichetele <body> </body> sau <frameset> </frameset>

(\*\*\*, 2018, )

#### CSS

CSS (Cascading Style Sheets) este un standard pentru formatarea elementelor unui document HTML. Stilurile se pot atașa elementelor HTML prin intermediul unor fișiere externe sau în cadrul documentului, prin elementul <style> și/sau atributul style. CSS se poate utiliza și pentru formatarea elementelor XHTML, XML și SVGL.

CSS3 reprezintă un upgrade ce aduce câteva atribute noi și ajută la dezvoltarea noilor concepte in webdesign.

Unele dintre cele mai importante segmente (module) noi adăugate acestui standard pentru formatarea elementelor HTML aduc un plus considerabil in dezvoltarea activități webdesign.

Mai jos sunt prezente in listă cele mai importante modulele adăugate in CSS3:

* Selectors
* Box Model
* Backgrounds and Borders
* Image Values and Replaced Content
* Text Effects
* 2D/3D Transformations
* Animations
* Multiple Column Layout
* User Interface

Deși au apărut unele deficiente de compatibilitate intre browsere, majoritatea proprietăților CSS3 au fost implementate cu succes in variantele browserelor noi.

(\*\*\*, 2018, )

#### JavaScript

JavaScript (JS) este un limbaj de programare orientat obiect bazat pe conceptul prototipurilor. Este folosit mai ales pentru introducerea unor funcționalități în paginile web, codul JavaScript din aceste pagini fiind rulat de către browser. Limbajul este binecunoscut pentru folosirea sa în construirea siturilor web, dar este folosit și pentru accesul la obiecte încastrate (embedded objects) în alte aplicații. A fost dezvoltat inițial de către Brendan Eich de la Netscape Communications Corporation sub numele de Mocha, apoi LiveScript, și denumit în final JavaScript.

În ciuda numelui și a unor similarități în sintaxă, între JavaScript și limbajul Java nu există nicio legătură. Ca și Java, JavaScript are o sintaxă apropiată de cea a limbajului C, dar are mai multe în comun cu limbajul Self decât cu Java.

Cea mai des întâlnită utilizare a JavaScript este în scriptarea paginilor web. Programatorii web pot îngloba în paginile HTML script-uri pentru diverse activități cum ar fi verificarea datelor introduse de utilizatori sau crearea de meniuri și alte efecte animate.

Browserele rețin în memorie o reprezentare a unei pagini web sub forma unui arbore de obiecte și pun la dispoziție aceste obiecte script-urilor JavaScript, care le pot citi și manipula. Arborele de obiecte poartă numele de Document Object Model sau DOM. Există un standard W3C pentru DOM-ul pe care trebuie să îl pună la dispoziție un browser, ceea ce oferă premiza scrierii de script-uri portabile, care să funcționeze pe toate browserele. În practică, însă, standardul W3C pentru DOM este incomplet implementat. Deși tendința browserelor este de a se alinia standardului W3C, unele din acestea încă prezintă incompatibilități majore, cum este cazul Internet Explorer.

O tehnică de construire a paginilor web tot mai întâlnită în ultimul timp este AJAX, abreviere de la „Asynchronous JavaScript and XML”. Această tehnică constă în executarea de cereri HTTP în fundal, fără a reîncărca toată pagina web, și actualizarea numai anumitor porțiuni ale paginii prin manipularea DOM-ului paginii. Tehnica AJAX permite construirea unor interfețe web cu timp de răspuns mic, întrucît operația (costisitoare ca timp) de încărcare a unei pagini HTML complete este în mare parte eliminată.

(\*\*\*, 2018, )

#### C#

Limbajul C# este unul dintre cele mai utilizate limbaje de programare multiparadigmă din lume, fiind clasat în acest moment pe locul 5 mondial, ca nivel de popularitate. Este un limbaj simplu, modern, cu o flexibilitate foarte mare în ceea ce privește dezvoltarea de aplicații și portabilitatea acestora (deși există unele voci care spun că portabilitatea este limitată, vom demonstra în continuare lipsa de veridicitate a acestei afirmații).

Istoria acestui limbaj își are originile la începutul mileniului în care ne aflăm. În iulie 2000 a fost lansată prima distribuție a limbajului C#, deși zvonurile apariției unui limbaj puternic au apărut încă de prin anul precedent. Microsoft a fost corporația care a decis să intre mult mai în forță în lumea dezvoltatorilor, întrucât limbajul Basic își pierduse popularitatea ceea ce a dus la o ușoară scădere a nivelului economic al companiei.

Unul dintre principalele motive care au inițiat crearea acestui limbaj de programare a fost legat de faptul că în anul 1995, cei de la Sun Microsystems au creat limbajul Java, primul limbaj multiparadigmă cu o portabilitate foarte mare. A înregistrat un nivel de popularitate exponențial în primii ani de la lansare, și chiar în zilele noastre, este pe primul loc mondial în ceea ce privește popularitatea. Astfel, Microsoft a decis să creeze un limbaj care să ofere mult mai multe elemente dezvoltatorilor.

Printre principalele calități ale limbajului, putem distinge:

* modernitate, simplitate, utilitate generală, productivitate mare
* stabilitate în cadrul aplicațiilor complexe, durabilitate
* este un limbaj total orientat pe obiect (orice entitate din acest limbaj este de fapt un obiect)
* oferă suport complet pentru dezvoltarea de componente necesare în medii distribuite, deci este și un limbaj
* orientat către componente

Limbajul C# are aplicabilitate foarte mare în cadrul sistemelor embeded, care includ dispozitive precum mp3 playere, semafoare, automate (vending machines), controlere, telefoane mobile, ceasuri digitale, s.a.m.d.

Deci acum știți că atunci când sunteți opriți la semafor, un algoritm transpus în C# coordonează controller-ul care dirijază fluxul de trafic.

(\*\*\*, 2018, )

#### SQL

SQL (de la numele englez Structured Query Language - limbaj de interogare structurat) este un limbaj de programare specific pentru manipularea datelor în sistemele de manipulare a bazelor de date relaționale (RDBMS), iar la origine este un limbaj bazat pe algebra relațională. Acesta are ca scop inserarea datelor, interogații, actualizare și ștergere, modificarea și crearea schemelor, precum și controlul accesului la date. A devenit un standard în domeniu (standardizat ANSI-ISO), fiind cel mai popular limbaj utilizat pentru crearea, modificarea, regăsirea și manipularea datelor de către SGBD-urile (Sistemele de Gestiune a Bazelor de Date) relaționale. Pe lângă versiunile standardizate ale limbajului, există o mulțime de dialecte și variante, unele proprietare, fiind specifice anumitor SGBD-uri și de asemenea conținând extensii pentru a suporta SBD-urile (Sistemele de Baze de Date) obiectuale (obiectual-relaționale).

SQL permite atât accesul la conținutul bazelor de date, cât și la structura acestora.

Limbajul SQL este divizat în următoarele elemente:

* Clauze, care sunt componente ale instrucțiunilor și interogărilor.
* Expresii, al căror efect este producerea de valori scalare sau tabele.
* Predicate, pot specifica condiții care sunt evaluate de SQL conform logicii ternare sau logicii booleene, în scopul limitării efectelor instrucțiunilor, sau pentru a influența cursul programului.
* Interogările, au ca scop regăsirea datelor după criterii specifice.
* Instrucțiunile, pot avea un efect persistent asupra datelor sau structurii datelor, sau pot controla tranzacțiile, conexiunile sau cursul programului. În general, instrucțiunile SQL se termină cu caracterul punct-virgulă (";"), deși acest lucru nu este obligatoriu în toate platformele SQL. Spațiile albe suplimentare sunt ignorate, dar ele pot fi folosite pentru lizibilitatea codului SQL.

(\*\*\*, 2018, )

### Tehnologii web

#### ASP.NET

ASP.NET este o tehnologie Microsoft pentru crearea de aplicații web și servicii web. ASP.NET este succesorul lui ASP (Active Server Pages) și beneficiază de puterea platformei de dezvoltare .NET, și de setul de instrumente oferite de mediul de dezvoltarea al aplicației „Visual Studio .NET”.

Cateva dintre avantajele ASP .NET sunt:

* ASP .NET are un set larg de componente, bazate pe XML, oferind astfel un model de programare orientat obiect (OOP).
* ASP .NET ruleaza cod compilat, ceea ce crește performanțele aplictiei web. Codul sursa poate fi separat în două fișiere, unul pentru codul executabil, iar un altul pentru continutul paginii (codul HTML și textul din pagină) .
* .NET este compatibil cu peste 20 de limbaje diferite, cele mai utilizate fiind C# si Visual Basic.

(\*\*\*, 2018, )

#### Entity Framework

Entity Framework este un set de tehnologii in ADO.NET ce suporta dezvoltarea de aplicatii software cu baze de date, aplicatii orientate pe obiecte. Comenzile din ADO.NET lucreaza cu scalari (date la nivel de coloana dintr-o tabela) in timp ce ADO.NET Entity Framework lucreaza cu obiecte (din baza de date se returneaza obiecte).Arhitectura ADO.NET Entity Framework consta din urmatoarele:

* Provideri specifici pentru sursa de date (Data source) ce abstractizeaza interfetele ADO.NET pentru conectare la baza de date cand programam folosind schema conceptuala (model conceptual).
* Provider specific bazei de date ce translateaza comenzile Entity SQL in cereri native SQL (comenzi SQL din limbajul de definire a bazei de date, limbajul de cereri).
* Parser EDM (Entity Data Model) si mapare vizualizari prin tratarea specificatiilor SDL (Storage Data Language – model de memorare) al modelului de date, stabilirea asociatiilor dintre modelul relational (baza de date) si modelul conceptual. Din schema relationala se creaza vizualizari ale datelor ce corespund modelului conceptual. Informatii din mai multe tabele sunt agregate intr-o entitate. Actualizarea bazei de date (apel metoda SaveChanges()) are ca efect construirea comenzilor SQL specifice fiecarei tabele ce apare in acea entitate.
* Servicii pentru metadata ce gestioneaza metadata entitatilor, relatiilor si maparilor.
* Tranzactii – pentru a suporta posibilitatile tranzactionale ale bazei de date.
* Utilitare pentru proiectare – Mapping Designer – incluse in mediul de dezvoltare.
* API pentru nivelul conceptual – runtime ce expune modelul de programare pentru a scrie cod folosind nivelul conceptual. Se folosesc obiecte de tip Connection, Command asemanator cu ADO.NET si se returneaza rezultate de tip EntityResultSets sau EntitySets.
* Componente deconectate – realizeaza un cache local pentru dataset si multimile entitati.
* Nivel de programare – ce expune EDM ca o constructie programabila si poate fi folosita in limbajele de programare ceea ce inseamna generare automata de cod pentru clasele CLR ce expun aceleasi proprietati ca o entitate, permitand instantierea entitatilor ca obiecte .NET sau expun entitatile ca servicii web.Modele de programare:
* Aplicatie centrata pe baza de date - Baza de date este in centrul aplicatiei.
* Aplicatie centrata pe model – modelul este in centrul aplicatiei. Accesul la date este facut pe baza modelului conceptual, model ce reflecta obiectele problemei de rezolvat. In acest caz exista posibilitatea de a folosi : Code first sau Model design first.

Modelul de programare centrat pe baza de date, presupune ca baza de date este creata si apoi se genereaza modelul logic ce contine tipurile folosite in logica aplicatiei. Acest lucru se face folosind mediul de dezvoltare VStudio. Se genereaza clasele POCO (EF versiune mai mare ca 4 si VS 2012/VS 2013) si fisierele necesare pentru nivelul conceptual, nivelul de mapare si nivelul de memorare.Aplicatia centrata pe model poate fi dezvoltata alegand una din variantele Code first sau Model design first.In cazul Code First, dezvoltatorul scrie toate clasele (POCO) modelului si clasa derivata din DbContext cu toate entitatile necesare si apoi cu ajutorul mediului de dezvoltare se creaza si genereaza baza de date, tabelele din baza de date si informatiile aditionale necesare pentru EF.

In cazul Model Design First, mediul de dezvoltare permite dezvoltarea unei diagrame a modelului aplicatiei si pe baza acesteia se va crea si genera baza de date, tabelele din baza de date si informatiile aditionale necesare pentru EF.

ADO.NET clasic presupune obiecte DataReader / DataAdapter pentru a citi informatii din baza de date si obiecte Command pentru a executa insert, update, delete, etc. DataAdapter din ADO .NET este folosit pentru DataSet si nu are echivalent in Entity Framework.

In EF randurile si coloanele din tabele sunt returnate ca obiecte si nu se foloseste in mod direct Command, se translateaza datele din forma tabelara in obiecte.

Obiecte de tip Command pot fi folosite pentru accesul direct la baza de date sau pentru cazurile pe care EF nu le poate rezolva.

EF foloseste un model numit Entity Data Model (EDM), dezvoltat din Entity Relationship Modeling (ERM).

Conceptele principale introduse de EDM sunt:

* Entity: entitatile sunt instante ale tipului Entity (de exemplu Customer, Order). Acestea reprezinta structura unei inregistrari identificata printr-o cheie. Entitatile sunt grupate in multimi de Entity (Entity-Sets).
* Relationship: relatiile asociaza entitatile si sunt instante ale tipurilor Relationship (de exemplu Order postat de Customer). Relatiile sunt grupate in multimi de relatii – Relationship-Sets.

EDM suporta diverse constructii ce extind aceste concepte de entitate si relatie:

* Inheritance: tipurile entitate pot fi definite astfel incat sa fie derivate din alte tipuri. Mostenirea in acest caz este una structurala, adica nu se mosteneste comportarea ci numai structura tipului entitate de baza. In plus la aceasta mostenire a structurii, o instanta a tipului entitate derivat satisface relatia “is a”.
* Tipuri complexe: EDM suporta definirea tipurilor complexe si folosirea acestora ca membri ai tipurilor entitate. De exemplu se poate defini tipul (clasa) Address ce are proprietatile Street, City si Telephone si apoi sa folosim o proprietate de tip Address in tipul entitate Customer.

ERM defineste o schema a entitatilor si a relatiilor dintre acestea.

Entitatile definesc schema unui obiect, dar nu si comportarea acestuia. Entitatea este asemanatoare cu schema unei tabele din baza de date numai ca aceasta descrie schema obiectelor problemei de rezolvat - pe scurt modelul.

EDM este un model pe partea de client si constituie fundamentul pentru EF.

EDM cuprinde trei niveluri, care sunt independente :

* nivelul conceptual (sau model conceptual/schema conceptuala) ;
* nivelul de mapare (sau model de mapare);
* nivelul de memorare (sau model de memorare).

Clasele entitate sunt continute in modelul conceptual din EDM.

Nivelul conceptual poate fi modelat prin scriere directa de cod (model de programare code first) – modelul este in centrul aplicatiei – sau folosind un utilitar pentru generarea entitatilor in situatia cand avem baza de date proiectata (model de programare database first). Sintaxa pentru nivelul conceptual este definita de Conceptual Schema Definition Language (CSDL).

Nivelul de memorare din EDM defineste tabele, coloane, relatii si tipuri de date ce sunt mapate la baza de date. Sintaxa pentru modelul de memorare este definita de Store Schema Definition Language (SSDL).

Nivelul de mapare defineste maparea (legatura) dintre nivelul conceptual si nivelul de memorare. Printre altele, acest nivel defineste cum sunt mapate proprietatile din clasele entitate la coloanele tabelelor din baza de date. Mapping Specification Language (MSL) defineste sintaxa pentru nivelul de memorare.

In EF, EDM este reprezentat in timpul proiectarii de un singur fisier XML din care la runtime se creeaza trei fisiere cu extensiile : csdl, msl, ssdl. Aceste fisiere contin descrierile pentru modelul conceptual, modelul de mapare si modelul de memorare.

(\*\*\*, 2018, )

#### Bootstrap

Bootstrap nu este un limbaj de programare care va dispărea peste 1-2 ani, ci este un mod ordonat de a utiliza elemente CSS, HTML şi Java Script pentru a crea structuri care se pot refolosi cu un simplu copy-paste.

În ansamblu poate economisi cu până la 70% din timpul de dezvoltare pe front-end pentru majoritatea proiectelor, de la landing-page-uri, la site-uri de prezentare şi magazine online.

Denumit iniţial Twitter Blueprint, framework-ul a fost dezvoltat de Mark Otto şi Jacob Thornton ca o soluţie internă pentru Twitter.

După câteva luni de la implementare, mai mulţi dezvoltatori au început să contribuie la el în timpul Hack Week, o săptămână în care angajaţii dezvoltau diverse proiecte, iar în august 2011 este lansat open-source sub denumirea de Bootstrap.

Bootstrap împarte pagina web în 12 coloane cu valori procentual-egale. Unind aceste coloane se pot obţine box-urile dorite pentru a adăuga conţinut.

Majoritatea site-urilor moderne utilizează aceste structuri, Bootstrap fiind de altfel acuzat că duce la realizarea de site-uri mult prea asemănătoare.

Este însă extrem de flexibil, iar clasele responsive pot fi controlate în funcţie de lăţimea dispozitivului.

Bootstrap pune la dispoziţie o serie de componente şi elemente des întâlnite: meniul de navigare cu dropdown, carouselul, breadcrumbs şi multe altele.

Pe lângă sistemul de griduri care îţi poate uşura semnificativ munca, Bootstrap vine şi cu o serie de stiluri pentru elementele de bază: butoane, formulare, tabele, imagini şi altele. Poate fi inclus foarte uşor în proiecte.

Bootstrap vine cu o documentare foarte bună din care se poate învăţa uşor.

Majoritatea template-urilor şi site-urilor noi utilizează Bootstrap în structura lor. Timpul de dezvoltare mai mic dar şi modificările care pot fi realizate cu uşurinţă îl fac să fie cerut.

Bootstrap poate salva un timp considerabil în dezvoltarea proiectelor. În plus este redat foarte bine în majoritatea dispozitivelor şi browserelor. Poate fi utilizat în proiectele de zi cu zi, de la landing page-uri şi site-uri de prezentare până la magazine online.

(\*\*\*, 2018, )

#### jQuery

jQuery este o platformă de dezvoltare JavaScript, concepută pentru a ușura și îmbunătăți procese precum traversarea arborelui DOM în HTML, managementul inter-browser al evenimentelor, animații și cereri tip AJAX. jQuery a fost gândit să fie cât mai mic posibil, disponibil în toate versiunile de browsere importante existente, și să respecte filosofia "Unobtrusive JavaScript". Biblioteca a fost lansată in 2006 de către John Resig.

jQuery se poate folosi pentru a rezolva următoarele probleme specifice programării web:

* selecții de elemente în arborele DOM folosind propriul motor de selecții open source Sizzle, un proiect născut din jQuery
* parcurgere și modificarea arborelui DOM (incluzând suport pentru selectori CSS 3 și XPath simpli)
* înregistrarea și modificarea evenimentelor din browser
* manipularea elementelor CSS
* efecte și animații
* cereri tip AJAX
* extensii ( vezi mai jos )
* utilităti - versiunea browser-ului, funcția each

Plugin-urile sau extensiile sunt unele dintre cele mai interesante aspecte ale jQuery. Arhitectura sa permite programatorilor să dezvolte subaplicații bazate în biblioteca principală care extind funcțiile de bază jQuery cu funcții specifice plugin-ului. În acest fel biblioteca principală poate ocupa foarte puțin spațiu, iar extensiile necesare în anumite pagini web pot fi încarcate la cerere, doar când este nevoie de ele. Există un set de extensii principal numit jQuery UI( jQuery User Interface). jQuery UI ofera un set de extensii pentru interactivitate de bază, efecte mai complexe decât cele din biblioteca de bază și teme de culori. Avantajul jQuery UI față de alte extensii este că dezvoltarea și testarea acestor componente se face în paralel cu dezvoltarea bibliotecii principale, minimizând riscul de incomptibilitate.

Orice programator poate crea o extensie și jQuery oferă publicare în catalogul de pe pagina proiectului în diversele categorii disponibile.

jQuery este un software liber, open-source licențiat sub licența MIT și GNU General Public License.

(\*\*\*, 2018, )

#### Ajax

Ajax (sau AJAX), prescurtare pentru Asynchronous JavaScript and XML, este o tehnică de programare pentru crearea de aplicații web interactive. Intenția este să facă paginile web să devină mai rapide și deci mai acceptate, prin schimbul în fundal al unor cantități mici de date cu serverul, astfel încât să nu fie nevoie ca pagina să fie reîncărcată la fiecare acțiune a utilizatorului. Aceasta are ca scop creșterea interactivității, vitezei și ușurinței în utilizare a aplicațiilor web.

Ajax nu este o tehnologie în sine. Termenul este folosit pentru definirea aplicațiilor web ce folosesc un ansamblu de tehnologii:

* HTML sau XHTML pentru structura semantică a informațiilor;
* CSS pentru prezentarea informațiilor;
* Javascript pentru interactivitate, pentru procesarea informațiilor prezentate;
* Obiectul XMLHttpRequest pentru schimbul și manipularea informațiilor într-o manieră asincronă cu server-ul web;
* XML este folosit de obicei pentru transferarea datelor între server și client, deși orice format funcționează, inclusiv HTML preformatat, text simplu etc.

(\*\*\*, 2018, )

#### LINQ

(\*\*\*, 2018, )

# Implementare

## Implementare hardware

## Implementare software

Implementarea aplicației software s-a realizat în mediul de programare Visual Studio cu limbajul de programare C#.

## Implementare firmware

Firmwareul a fost implementat în mediul integrat de dezvoltare Arduino, cu limbajul C.

# Interfața cu utilizatorul

## Dispozitivul de pontare

## Aplicatia web

# Concluzii, contribuţii şi direcţii de continuare a dezvoltării

Sistemele de pontaj capteaza interesul majoritatii firmelor, in contextul unei piete de munca dinamice si a controlului cheltuielilor salariale. Acestea reprezinta un instrument obiectiv de masura a timpului de lucru al angajatiilor.

Integrate cu sistemul de control acces, cu cel de control si planificare a productiei si a resurselor umane, toate aceste sisteme aduc un aport deosebit intr-un punct esential al managementului societatilor comerciale moderne, prin planificarea, controlul costurilor si motivarea personalului.

In cazul societatiilor mijlocii si mari, resursele umane si costurile aferente intocmirii pontajului sunt considerabile, fiind subiective si fara un control riguros, implementarea unui sistem de pontaj electronic fiind justificata.

Programele de pontaj asigură administrarea unui numar nelimitat de angajati, pontajul dupa programe si schimburi de lucru, inregistrarea orelor suplimentare si a intarzierilor, planificarea concediilor si a zilelor de sarbatoare legale, drepturi specifice de acces si pontaj, definirea de motive de absenta si de plata.

Interfata cu echipamentele de colectare a inregistrarilor de pontaj se face prin module software specifice, care asigura o comunicatie in timp real. Acest avantaj ofera personalului responsabil, o informare asupra prezentei angajatilor cu scopul unor decizii rapide. Aplicatile software permit generarea unui numar de rapoarte specifice cat si exportul datelor de pontaj.

# Bibliografie

\*\*\* (2018, ), Stadiul actual, <https://www.calculatorvenituri.ro/fisa-de-pontaj-obligatorie-cum-sa-tii-evidenta-angajatilor-cu-un-soft-de-pontaj>

\*\*\* (2018, ), FPC-4500, <http://www.yli.ro/control-acces-si-pontaj/sisteme-de-pontaj/colector-de-amprente-usb.html>

\*\*\* (2018, ), FPA-1700, <http://www.yli.ro/control-acces-si-pontaj/sisteme-de-pontaj/controler-de-acces-cu-functie-de-pontaj-cu-cititoare-de-amprente-si-de-cartele-incorporate.html>

\*\*\* (2018, ), FPA-1600, <http://www.yli.ro/control-acces-si-pontaj/sisteme-de-pontaj/controler-de-acces-cu-functie-de-pontaj-cu-cititoare-de-amprente-si-de-cartele-incorporate-1125.html>

\*\*\* (2018, ), PRO-CAPTURE-T, <http://www.yli.ro/control-acces-si-pontaj/sisteme-de-pontaj/controler-stand-alone-cu-functie-de-pontaj-cu-amprenta-card-si-tastatura-zk-bio-security.html>

\*\*\* (2018, ), SilkFP-101TA, <http://www.yli.ro/control-acces-si-pontaj/sisteme-de-pontaj/terminal-de-pontaj-cu-ecran-tft-3-5-senzor-amprenta-silkid.html>

\*\*\* (2018, ), Identificare biometrică, <http://perform.usv.ro/rapoarte/13/raport_cercetare_1.pdf>

\*\*\* (2018, ), Dactiloscopie, <https://ro.wikipedia.org/wiki/Dactiloscopie>

\*\*\* (2018, ), Conexiune Wi-Fi, <https://ro.wikipedia.org/wiki/Wi-Fi>

\*\*\* (2016, ), Programare orientată pe obiecte, <https://ro.wikipedia.org/wiki/Programare_orientat%C4%83_pe_obiecte>

\*\*\* (2018, ), Principiile SOLID, https://ctrl-d.ro/tips-and-tricks/programarea-orientata-pe-obiect-si-dezvoltarea-de-aplicatii-5-principii-de-design/

\*\*\* (2018, ), Modelul arhitectural MVC, <https://ro.wikipedia.org/wiki/Model-view-controller>

\*\*\* (2018, ), Dezvoltarea aplicațiilor web, <https://ro.wikipedia.org/wiki/Aplicație_web>

\*\*\* (2018, ), Controlul versiunilor, <https://ro.wikipedia.org/wiki/Controlul_versiunilor>

\*\*\* (2016, ), Imprimare 3D, <https://en.wikipedia.org/wiki/3D_printing>

\*\*\* (2016, ), Imprimare 3D, <http://www.printing3d.ro/avantaje/>

\*\*\* (2018, ), Arduino Mega 2560, https://store.arduino.cc/usa/arduino-mega-2560-rev3

\*\*\* (2018, ), Senzor de amprentă, <https://www.adafruit.com/product/751#tutorials>

\*\*\* (2018, ), 2.8" TFT Touchscreen Shield, <https://www.adafruit.com/product/1947#tutorials>

\*\*\* (2018, ), WiFi Shield, <https://www.adafruit.com/product/3653#tutorials>

\*\*\* (2018, ), Piezo, <https://www.adafruit.com/product/1740#tutorials>

\*\*\* (2018, ), Buton On/Off Switch, <http://www.adafruit.com/product/915#tutorials>

\*\*\* (2016, ), Arduino IDE, <https://www.arduino.cc/en/Guide/Environment>

\*\*\* (2018, ), Visual Studio, <https://ro.wikipedia.org/wiki/Microsoft_Visual_Studio>

\*\*\* (2018, ), Visual Studio Code, <https://www.mobzine.ro/2015/04/download-visual-studio-code-gratuit-pe-windows-macos-si-linux/>

\*\*\* (2018, ), Visual Studio Code, <https://www.bobses.eu/2015/11/20/microsoft-a-deschis-sursa-pentru-editorul-de-cod-visual-code-studio/>

\*\*\* (2018, ), SQL Server Management Studio, <https://en.wikipedia.org/wiki/SQL_Server_Management_Studio>

\*\*\* (2018, ), Microsoft SQL Server, <https://ro.wikipedia.org/wiki/Microsoft_SQL_Server>

\*\*\* (2018, ), Microsoft IIS, <https://ro.wikipedia.org/wiki/Internet_Information_Services>

\*\*\* (2016, ), Tinkercad, <https://www.tinkercad.com/>

\*\*\* (2016, ), C, <https://ro.wikipedia.org/wiki/C_(limbaj_de_programare)>

\*\*\* (2018, ), HTML, <https://ro.wikipedia.org/wiki/HyperText_Markup_Language>

\*\*\* (2018, ), CSS, <https://ro.wikipedia.org/wiki/Cascading_Style_Sheets>

\*\*\* (2018, ), JavaScript, <https://ro.wikipedia.org/wiki/JavaScript>

\*\*\* (2018, ), C#, <https://www.code-it.ro/tutoriale-c-prezentarea-limbajului/>

\*\*\* (2018, ), SQL, <https://ro.wikipedia.org/wiki/SQL>

\*\*\* (2018, ), ASP.NET, <https://ro.wikipedia.org/wiki/ASP.NET>

\*\*\* (2018, ), Entity Framework, <https://profs.info.uaic.ro/~iasimin/Special/Curs_EntityFramework.pdf>

\*\*\* (2018, ), Bootstrap, <https://ctrl-d.ro/tutoriale/ce-este-bootstrap-si-cum-te-poate-ajuta-in-dezvoltarea-de-site-uri-web-moderne/>

\*\*\* (2018, ), jQuery, <https://ro.wikipedia.org/wiki/JQuery>

\*\*\* (2018, ), Ajax, <https://ro.wikipedia.org/wiki/Ajax_(programare)>