Implementarea unui sistem avansat de control al accesului pe baza unui abonament

Student:

Ciobanu Laurențiu Ștefan

# CUPRINS

[CUPRINS 2](#_Toc1783646347)

1.Introducere 4

[1.1 Scopul documentului 4](#_Toc1659812674)

[2.Prezentare generală și abordări de proiectare 5](#_Toc877488453)

[2.1 Prezentare generală 5](#_Toc1328452156)

[2.2 Presupuneri/ Constrângeri/ Riscuri 5](#_Toc623510857)

[2.2.1Presupuneri 5](#_Toc1566339516)

[2.2.2Constrângeri 5](#_Toc1884209966)

[2.2.3Riscuri 5](#_Toc1622584899)

[3.Considerațiii de proiectare 6](#_Toc2061953731)

[3.1 Obiective și linii directoare (ghiduri) 6](#_Toc664845161)

[3.2 Metode de dezvoltare 6](#_Toc68397684)

[3.3 Strategii de arhitectură 7](#_Toc575468306)

[4.Arhitectura Sistemului și Proiectarea Arhitecturii 8](#_Toc1677799952)

[4.1 Vedere logică 8](#_Toc1647584150)

[4.2 Arhitectură hardware 8](#_Toc36652913)

[4.3 Arhitectură software 8](#_Toc91559432)

[4.4 Arhitectura informațiilor 9](#_Toc1238263487)

[4.5 Arhitectura de comunicații interne 9](#_Toc1225397083)

[4.6 Diagrama de arhitectură a sistemului 9](#_Toc934312667)

[5.Proiectarea sistemului 9](#_Toc598949626)

[5.1 Proiectarea bazei de date 9](#_Toc648700992)

[5.1.1Obiecte de date și structuri de date rezultante 9](#_Toc428217135)

[5.1.2Fișiere și baze de date 10](#_Toc1242161070)

[5.2 Conversii de date 10](#_Toc1676430599)

[5.3 Interfețe utilizator 10](#_Toc2133732553)

[5.3.1Intrări 10](#_Toc1545620057)

[5.3.2Ieșiri 10](#_Toc1510652309)

[5.4 Proiectarea interfețelor cu utilizatorul 11](#_Toc1668060100)

[6.Scenarii de utilizare 11](#_Toc1814866918)

[7.Proiectare de detaliu 12](#_Toc1876992510)

[7.1 Proiectare hardware de detaliu 12](#_Toc1051052986)

[7.2 Proiectare software de detaliu 13](#_Toc1757473899)

[7.3 Proiectare detaliată de securitate 13](#_Toc1628531879)

[7.4 Proiectare de detaliu pentru performanța sistemului 13](#_Toc1107055851)

[7.5 Proiectare detaliată a comunicațiilor interne (între componente) 13](#_Toc1845631458)

[8.Controale pentru verificarea integrității sistemului 14](#_Toc665986166)

**1.** **Introducere**

Numele Proiectului: Implementarea unui Sistem Avansat de Control al Accesului pe Baza unui Abonament  
Acronim: ISCAA  
 Evoluția Documentului: Acest Document de Proiectare a Sistemului marchează o etapă crucială în procesul de dezvoltare a sistemului ISCAA. Transformă cerințele funcționale și non-funcționale definite în Documentul de Cerințe în specificații de proiectare mai tehnice, care vor servi drept fundament pentru construirea și implementarea sistemului.  
 Considerații Referitoare la Securitate și Confidențialitate: Datorită naturii datelor sensibile implicate în gestionarea accesului, se acordă o atenție specială securității și confidențialității. Acest document va include specificații și măsuri de securitate pentru protejarea informațiilor personale și a datelor de acces.  
 Descrierea Proiectului: Documentul de Proiectare a Sistemului va detalia modul în care cerințele funcționale și non-funcționale sunt translate în specificații de proiectare a sistemului. Se va prezenta arhitectura sistemului la nivel înalt, proiectarea datelor, interfața utilizatorului și fluxurile operaționale. De asemenea, va fi documentată proiectarea detaliată a fiecărei componente a sistemului, inclusiv hardware-ul, software-ul și integrarea cu alte sisteme existente. Acest document va fi o resursă esențială pentru echipa de dezvoltare în timpul construirii și implementării sistemului ISCAA.

## 1.1 **Scopul documentului**

Scopul proiectului "Implementarea unui Sistem Avansat de Control al Accesului pe Baza unui Abonament" este de a dezvolta și implementa o soluție modernă și eficientă pentru gestionarea accesului în diverse medii, în concordanță cu evoluția rapidă a tehnologiei și cu creșterea preocupării pentru securitate. Scopul este de a asigura protecția adecvată a persoanelor și a bunurilor prin utilizarea unui sistem avansat de control al accesului, bazat pe un model de abonament.

# **2.** **Prezentare generală și abordări de proiectare**

2.1 **Prezentare generală**

Proiectul constă în implementarea unui sistem inovator de control al accesului bazat pe abonament, ce utilizează tehnologia avansată de citire a amprentei. Acest sistem oferă utilizatorilor acces la zone sau servicii specifice în funcție de abonamentul lor și de autentificarea lor prin amprentă. Conceput pentru a satisface nevoile de securitate și accesibilitate din diverse medii, inclusiv clădiri de birouri, instituții guvernamentale și facilități comerciale, proiectul se axează pe eficiență și fiabilitate.

2.2 **Presupuneri/ Constrângeri/ Riscuri**

2.2.1 **Presupuneri**

Sistemul automatizat funcționează corect în conformitate cu descrierea dată, având capacitatea de a detecta și de a reacționa la intrările și ieșirile utilizatorilor.

Utilizatorii sunt familiarizați cu utilizarea sistemului și își folosesc abonamentele corespunzător.

Adminul platformei are acces la sistem pentru a gestiona evenimentele neașteptate și pentru a rezolva problemele.

## 2.2.2 **Constrângeri**

Dependența de funcționarea corectă a senzorilor ultrasonici și a senzorului de amprentă pentru detectarea și autentificarea utilizatorilor.

Necesitatea ca adminul să fie disponibil pentru a rezolva problemele sau pentru a gestiona situațiile neașteptate.

Do youLimitările tehnice și financiare pot influența dezvoltarea și implementarea sistemului.

## 2.2.3 **Riscuri**

Eșecul senzorilor sau a altor componente tehnice poate duce la întreruperi în funcționarea sistemului și la o scădere a securității.

Utilizatorii care încearcă să abuzeze de sistem, cum ar fi încercarea de a intra cu mai multe persoane pe un singur abonament, pot afecta securitatea și integritatea sistemului.

Situații neașteptate, cum ar fi nefuncționarea senzorului de amprentă sau pierderea conexiunii la platforma adminului, pot crea dificultăți în gestionarea sistemului și în asigurarea accesului corect.

**3.** **Considerațiii de proiectare**

3.1 **Obiective și linii directoare (ghiduri)**

1. Creșterea securității: Implementarea unui sistem avansat de control al accesului are ca obiectiv principal sporirea nivelului de securitate în interiorul unui spațiu.

2. Controlul accesului: Un alt obiectiv important este asigurarea unui control riguros asupra accesului în diverse zone sau facilități.

3. Eficiența operațională: Implementarea unui sistem de control al accesului pe baza unui abonament poate aduce beneficii semnificative în ceea ce privește eficiența operațională.

4. Reduzerea costurilor: Un obiectiv important poate fi și reducerea costurilor asociate cu implementarea și întreținerea sistemului de control al accesului.

5. Integrarea cu alte tehnologii: Implementarea unui sistem avansat de control al accesului ar trebui să permită integrarea cu alte tehnologii moderne, cum ar fi sistemele de supraveghere video, recunoașterea facială sau scanarea biometrică.

6. Asigurarea conformității: Un alt obiectiv important poate fi asigurarea conformității cu reglementările

3.2 **Metode de dezvoltare**

Integrarea plăcii Arduino Mega cu celelalte componente hardware și software necesare pentru funcționarea sistemului de control al accesului. Extinderea capacităților sistemului prin adăugarea de module suplimentare, cum ar fi modulele de comunicație sau alte senzori avansați, pentru a îmbunătăți funcționalitățile și caracteristicile sistemului.

Documentarea completă a configurației hardware și a codului software pentru a facilita înțelegerea și gestionarea sistemului. Furnizarea instrucțiunilor de utilizare și depanare pentru adminul sistemului și utilizatorii finali. Asigurarea unei mentenanțe periodice pentru a menține funcționalitatea și securitatea sistemului pe termen lung.

## 3.3 **Strategii de arhitectură**

1. Arhitectură bazată pe microservicii:

Utilizarea unei arhitecturi bazate pe microservicii permite dezvoltarea și scalarea independentă a diferitelor componente ale sistemului, precum autentificarea, gestionarea abonamentelor și controlul accesului.

Fiecare serviciu poate fi dezvoltat și implementat separat, permițând o gestionare mai eficientă a codului și a resurselor.

2. Arhitectură orientată pe evenimente:

Adoptarea unei arhitecturi orientate pe evenimente facilitează gestionarea fluxurilor de evenimente generate de sistemele de detectare și autentificare, precum și de alte acțiuni ale utilizatorilor.

Utilizarea unui broker de mesaje sau a unui sistem de cozi de evenimente poate asigura trimiterea și prelucrarea eficientă a evenimentelor în timp real.

3. Utilizarea unui framework de securitate robust:

Integrarea unui framework de securitate robust, cum ar fi Spring Security în Java sau Django Authentication în Python, poate asigura autentificarea și autorizarea corespunzătoare a utilizatorilor și a accesului la resursele sistemului.

Implementarea unor politici de securitate stricte și a mecanismelor de criptare poate contribui la protejarea datelor sensibile și la prevenirea accesului neautorizat.

4. Arhitectură modulară și flexibilă:

Dezvoltarea unei arhitecturi modulare și flexibile permite integrarea ușoară a diferitelor componente hardware și software, precum și extinderea sistemului în viitor cu noi funcționalități sau tehnologii.

Utilizarea unor standarde deschise și a interfețelor bine definite facilitează interoperabilitatea și compatibilitatea cu alte sisteme și dispozitive.

5. Utilizarea tehnologiilor IoT (Internet of Things):

Integrarea tehnologiilor IoT, precum senzori ultrasonici sau dispozitive de recunoaștere facială, poate îmbunătăți eficiența și precizia sistemului de control al accesului.

Utilizarea unor protocoale de comunicație standard, cum ar fi MQTT sau CoAP, poate facilita comunicarea între diferitele dispozitive și componente ale sistemului.

# **4.** **Arhitectura Sistemului și Proiectarea Arhitecturii**

## 4.1 **Vedere logică**

Utilizarea plăcii Arduino Mega ca unitate centrală de control, responsabilă pentru gestionarea tuturor componentelor sistemului.

Conectarea senzorului ultrasonic pentru detectarea intrărilor și ieșirilor utilizatorilor.

Integrarea unui senzor de amprentă pentru autentificarea utilizatorilor.

Controlul unui servomotor pentru deschiderea și închiderea ușilor în funcție de autentificare și acces.

Utilizarea butonului pentru diverse interacțiuni manuale, cum ar fi înregistrarea unei noi amprente sau acționarea unui buton de panică.

## 4.2 **Arhitectură hardware**

Placa Arduino Mega servește ca nucleu al sistemului, gestionând toate conexiunile și funcțiile componente.

Senzorul ultrasonic este conectat la placa Arduino pentru detectarea intrărilor și ieșirilor utilizatorilor.

Senzorul de amprentă este conectat la placa Arduino pentru autentificarea utilizatorilor.

Servomotorul este conectat la placa Arduino pentru acționarea ușilor.

Celelalte componente, cum ar fi LED-urile, buzzerul, butonul și rezistențele, sunt integrate în sistem pentru diverse scopuri, cum ar fi semnalizarea stării sistemului sau interacțiunea utilizatorului.

## 4.3 **Arhitectură software**

Dezvoltarea codului software pentru placa Arduino Mega în limbajul de programare Arduino, utilizând mediul de dezvoltare Arduino IDE.

Implementarea algoritmilor de control al accesului, inclusiv detectarea și procesarea semnalelor de la senzorul ultrasonic și senzorul de amprentă, precum și acționarea servomotorului pentru deschiderea și închiderea ușilor.

Gestionarea interacțiunilor cu utilizatorul prin intermediul butonului și afișarea stării sistemului folosind LED-urile și buzzerul.

## 4.4 **Arhitectura informațiilor**

Utilizarea unei structuri de date pentru gestionarea informațiilor despre utilizatori și abonamente, inclusiv datele de identificare și privilegiile de acces asociate.

Stocarea și gestionarea informațiilor de autentificare, cum ar fi șabloanele de amprente, într-o formă sigură și protejată.

## 4.5 **Arhitectura de comunicații interne**

Comunicarea internă între componentele hardware și software se realizează prin intermediul conexiunilor fizice între placa Arduino Mega și celelalte componente hardware.

Comunicarea între diferitele părți ale codului software se face prin intermediul variabilelor și funcțiilor definite în cadrul programului Arduino.

## 4.6 **Diagrama de arhitectură a sistemului**

Elaborarea unei diagrame de arhitectură a sistemului, care să prezinte interacțiunile și fluxurile de date între diferitele componente și module ale sistemului.

Aceste propuneri ar trebui să ofere o abordare detaliată și bine structurată a arhitecturii proiectului, luând în considerare componentele furnizate și obiectivele sistemului.

**5.** **Proiectarea sistemului**

## 5.1 **Proiectarea bazei de date**

## 5.1.1 **Obiecte de date și structuri de date rezultante**

Tabelul "Utilizatori" pentru stocarea informațiilor despre utilizatori, inclusiv ID-ul utilizatorului, numele, prenumele, abonamentul asociat și alte detalii relevante.

Tabelul "Amprente" pentru înregistrarea amprentelor utilizatorilor, cu referințe către ID-urile utilizatorilor din tabelul "Utilizatori".

Tabelul "Acces" pentru înregistrarea evenimentelor de acces, precum data și ora accesului, tipul de acces (intrare/ieșire) și ID-ul utilizatorului asociat.

## 5.1.2 **Fișiere și baze de date**

Utilizarea unei baze de date relationale, cum ar fi MySQL sau SQLite, pentru a stoca și gestiona datele sistemului.

Folosirea unui mediu de dezvoltare integrat, cum ar fi phpMyAdmin, pentru administrarea și gestionarea bazei de date.

## 5.2 **Conversii de date**

Implementarea funcțiilor pentru conversia datelor între diferite formate, cum ar fi conversia datelor de la senzorii hardware la formatul necesar pentru stocare în baza de date sau conversia datelor de la baza de date la formatul necesar pentru afișare în interfețele cu utilizatorul.

## 5.3 **Interfețe utilizator**

## 5.3.1 **Intrări**

Dezvoltarea unei interfețe de utilizator pentru înregistrarea și gestionarea utilizatorilor și a abonamentelor lor, inclusiv funcționalități pentru adăugare, modificare și ștergere de utilizatori și abonamente.

Crearea unei interfețe pentru înregistrarea amprentelor utilizatorilor în sistem.

## 5.3.2 **Ieșiri**

Realizarea unei interfețe pentru afișarea evenimentelor de acces înregistrate în baza de date, precum și pentru monitorizarea stării sistemului în timp real.

Implementarea unor alerte și notificări vizuale sau auditive pentru semnalizarea evenimentelor sau a situațiilor neobișnuite, cum ar fi încercările de acces neautorizat sau problemele hardware.

5.4 **Proiectarea interfețelor cu utilizatorul**

Utilizarea unui design intuitiv și prietenos pentru utilizator în cadrul interfețelor cu utilizatorul, pentru a facilita utilizarea și înțelegerea funcționalităților sistemului.

Asigurarea unei interfețe corespunzătoare pentru gestionarea eficientă a datelor și a interacțiunilor utilizatorului cu sistemul, cu un accent deosebit pe ușurința înregistrării și gestionării utilizatorilor și a evenimentelor de acces.

Aceste propuneri ar trebui să ofere o viziune cuprinzătoare și bine definită asupra proiectării și dezvoltării bazei de date, a interfețelor cu utilizatorul și a altor aspecte relevante pentru proiectul "Implementarea unui sistem avansat de control al accesului pe baza unui abonament".

**6.** **Scenarii de utilizare**

Scenariu 1: Utilizatorul autorizat accesează spațiul:

Utilizatorul se apropie de ușa de acces și senzorul ultrasonic detectează prezența sa.

Sistemul activează senzorul de amprentă și solicită utilizatorului să își autentifice amprenta.

Utilizatorul își plasează amprenta pe senzorul de amprentă și este recunoscut de sistem.

LED-ul verde se aprinde, iar servomotorul deschide ușa pentru a permite accesul utilizatorului.

Utilizatorul intră în spațiu, iar senzorul ultrasonic detectează ieșirea sa.

Sistemul înregistrează evenimentul de acces și închide ușa automat după ieșirea utilizatorului.

Scenariu 2: Utilizatorul încearcă accesul fără autorizație:

Utilizatorul neautorizat se apropie de ușa de acces și senzorul ultrasonic detectează prezența sa.

Sistemul activează senzorul de amprentă și solicită utilizatorului să își autentifice amprenta.

Utilizatorul încearcă să folosească senzorul de amprentă, dar sistemul îl recunoaște ca neautorizat.

LED-ul roșu se aprinde, iar buzzerul emite un semnal sonor de avertisment.

Sistemul înregistrează încercarea de acces neautorizat și refuză deschiderea ușii.

Utilizatorul este îndrumat să contacteze adminul sistemului pentru soluționarea problemei.

Scenariu 3: Administrarea utilizatorilor și a abonamentelor:

Adminul sistemului accesează interfața de administrare și selectează opțiunea de gestionare a utilizatorilor.

Adminul adaugă un nou utilizator în sistem, introducând informațiile necesare, cum ar fi numele, prenumele și amprenta.

Sistemul înregistrează noul utilizator și îi atribuie un abonament corespunzător.

Adminul poate modifica sau șterge utilizatori existenți sau poate actualiza informațiile lor.

Adminul poate gestiona și abonamentele utilizatorilor, inclusiv crearea, modificarea sau anularea acestora, în funcție de necesități.

Aceste simulări oferă o imagine a modului în care proiectul poate fi utilizat în diferite situații și cum sistemul reacționează la diverse evenimente și interacțiuni cu utilizatorii și adminul sistemului.

# **7.** **Proiectare de detaliu**

## 7.1 **Proiectare hardware de detaliu**

Placa Arduino Mega: Se utilizează ca unitate centrală de control a sistemului, gestionând toate componentele hardware și interacțiunile cu acestea.

ESP8266: Conexiune Wi-Fi pentru posibilitatea de a accesa sistemul de la distanță și pentru actualizări de software.

Senzor ultrasonic: Pentru detectarea prezenței utilizatorilor în fața ușii de acces.

Senzor de amprentă: Pentru autentificarea utilizatorilor înainte de a li se permite accesul.

Servomotor: Pentru acționarea mecanică a ușii de acces.

Breadboard, buton, rezistențe, LED-uri, buzzer: Componente auxiliare pentru interacțiunea cu utilizatorul și semnalizarea stării sistemului.

Fire pentru conexiuni: Pentru interconectarea tuturor componentelor și pentru a asigura alimentarea și comunicațiile între acestea.

## 7.2 **Proiectare software de detaliu**

Dezvoltarea codului software pentru placa Arduino Mega, folosind limbajul de programare Arduino.

Implementarea algoritmilor de control al accesului, inclusiv gestionarea semnalelor de la senzorul ultrasonic și de la senzorul de amprentă, precum și controlul servomotorului pentru acționarea ușii.

Crearea interfețelor de utilizator pentru gestionarea utilizatorilor și a abonamentelor, inclusiv funcționalități de adăugare, modificare și ștergere.

Implementarea funcțiilor de securitate, cum ar fi criptarea datelor și autentificarea utilizatorilor.

7.3 **Proiectare detaliată de securitate**

Utilizarea criptării pentru protejarea datelor sensibile, cum ar fi informațiile despre utilizatori și amprentele.

Implementarea unor politici de securitate stricte pentru gestionarea accesului la sistem și pentru protejarea împotriva accesului neautorizat.

Actualizarea regulată a software-ului și a firmware-ului pentru a remedia vulnerabilitățile de securitate și pentru a menține sistemul în condiții sigure de funcționare.

## 7.4 **Proiectare de detaliu pentru performanța sistemului**

Optimizarea codului software pentru a asigura o performanță și o eficiență maximă a sistemului.

Utilizarea algoritmilor și a tehnologiilor eficiente pentru a minimiza timpul de răspuns al sistemului și pentru a asigura o funcționare rapidă și fluidă.

Testarea sistematică a performanței sistemului în diferite condiții și scenarii pentru a identifica și remedia eventualele probleme de performanță.

## 7.5 **Proiectare detaliată a comunicațiilor interne (între componente)**

Definirea și implementarea protocolului de comunicare între placa Arduino Mega și celelalte componente hardware.

Asigurarea unei comunicări fiabile și sigure între componentele sistemului pentru schimbul de date și comenzi.

Testarea și validarea comunicării interne pentru a asigura interoperabilitatea și fiabilitatea sistemului.

**8.** **Controale pentru verificarea integrității sistemului**

Pentru a implementa controale pentru verificarea integrității sistemului în cadrul proiectului, putem utiliza diverse tehnici și mecanisme pentru a asigura funcționarea corectă și sigură a sistemului. Iată câteva exemple:

**Monitorizarea stării hardware:**

Implementarea unui sistem de monitorizare a stării hardware pentru a detecta eventuale defecțiuni sau anomalii în funcționarea componentelor hardware, cum ar fi placa Arduino Mega, senzorul ultrasonic, senzorul de amprentă etc.

Utilizarea unor senzori suplimentari sau tehnologii de diagnosticare pentru a identifica temperaturi anormale, fluctuații de tensiune sau alte probleme ale componentelor hardware.

**Verificarea integrității software-ului:**

Implementarea unui sistem de verificare a integrității software-ului pentru a detecta eventuale modificări sau corupții în codul software al sistemului.

Utilizarea unor mecanisme de semnătură digitală sau hash-uri pentru a verifica autenticitatea și integritatea fișierelor de programare și a actualizărilor de software.

**Loguri și jurnale de evenimente:**

Crearea unui sistem de înregistrare și monitorizare a evenimentelor pentru a evidenția activitățile și interacțiunile sistemului cu utilizatorii și cu mediul înconjurător.

Analiza periodică a logurilor și jurnalelor de evenimente pentru a identifica potențiale probleme sau activități suspecte.

**Sisteme de backup și recuperare a datelor:**

Implementarea unui sistem de backup și recuperare a datelor pentru a asigura protecția și disponibilitatea informațiilor critice, cum ar fi baza de date a utilizatorilor și a amprentelor.

Realizarea unor copii de rezervă periodice și stocarea acestora într-un mediu sigur și protejat.

**Teste de securitate și penetrație:**

Efectuarea unor teste periodice de securitate și penetrație pentru a evalua vulnerabilitățile și riscurile sistemului și pentru a identifica posibile puncte slabe sau expuneri la atacuri.

Implementarea măsurilor corective și îmbunătățirea continuă a securității sistemului în funcție de rezultatele testelor.

**Actualizări și patch-uri de securitate:**

Menținerea sistemului actualizat cu cele mai recente patch-uri de securitate și actualizări de software pentru a remedia eventualele vulnerabilități și probleme de securitate cunoscute.

Implementarea unui proces de actualizare și gestionare a patch-urilor pentru a asigura securitatea și stabilitatea sistemului în timp.