Dezvoltarea unui sistem inteligent de control pentru o locuinta

# Documentul de proiectare

Student: Lazar Mihai-Valentin

Cuprins

[1. Introducere 1](#_Toc160527836)

[1.1 Scopul documentului 1](#_Toc160527837)

[2. Prezentare generală și abordări de proiectare 2](#_Toc160527838)

[2.1 Prezentare generală 2](#_Toc160527839)

[2.2 Presupuneri/ Constrângeri/ Riscuri 2](#_Toc160527840)

[2.2.1 Presupuneri 2](#_Toc160527841)

[2.2.2 Constrângeri 2](#_Toc160527842)

[2.2.3 Riscuri 3](#_Toc160527843)

[3. Considerațiii de proiectare 4](#_Toc160527844)

[3.1 Obiective și linii directoare (ghiduri) 4](#_Toc160527845)

[3.2 Metode de dezvoltare 4](#_Toc160527846)

[3.3 Strategii de arhitectură 4](#_Toc160527847)

[4. Arhitectura Sistemului și Proiectarea Arhitecturii 6](#_Toc160527848)

[4.1 Vedere logică 6](#_Toc160527849)

[4.2 Arhitectură hardware 6](#_Toc160527850)

[4.3 Arhitectură software 6](#_Toc160527851)

[4.4 Arhitectura informațiilor 7](#_Toc160527852)

[4.5 Arhitectura de comunicații interne 7](#_Toc160527853)

[4.6 Diagrama de arhitectură a sistemului 8](#_Toc160527854)

[5. Proiectarea sistemului 9](#_Toc160527855)

[5.1 Proiectarea bazei de date 9](#_Toc160527856)

[5.1.1 Obiecte de date și structuri de date rezultante 9](#_Toc160527857)

[5.1.2 Fișiere și baze de date 9](#_Toc160527858)

[5.2 Conversii de date 9](#_Toc160527859)

[5.3 Interfețe utilizator 10](#_Toc160527860)

[5.3.1 Intrări 10](#_Toc160527861)

[5.3.2 Ieșiri 10](#_Toc160527862)

[5.4 Proiectarea interfețelor cu utilizatorul 10](#_Toc160527863)

[6. Scenarii de utilizare 11](#_Toc160527864)

[7. Proiectare de detaliu 12](#_Toc160527865)

[7.1 Proiectare hardware de detaliu 12](#_Toc160527866)

[7.2 Proiectare software de deatliu 12](#_Toc160527867)

[7.3 Proiectare detaliată de securitate 13](#_Toc160527868)

[7.4 Proiectare de detaliu pentru performanța sistemului 14](#_Toc160527869)

[7.5 Proiectare detaliată a comunicațiilor interne (între componente) 14](#_Toc160527870)

[8. Controale pentru verificarea integrității sistemului 15](#_Toc160527871)

[Anexa A: Gestiunea modificărilor documentului 16](#_Toc160527872)

## Introducere

In prezentul document sunt prezentate detalii si instructiuni legate de realizarea proiectului de Dezvoltare a unui sistem inteligent de control pentru o locuinta, punandu-se accent pe proiectarea acestuia si cum a fost gandit sa functioneze, acesta incluzand cerintele hardware, software, dependentele externe si cerintele de securitate. Proiectarile si obiectivele curente propun starea finala a sistemului, putand a fi modificate in functie de evolutia sistemului.

### Scopul documentului

Scopul acestui document este de a ajuta si indruma echipa de dezvoltare si managerul de proiect in a pune in practica si a realiza cerintele de proietare ale proiectului. Se vor documenta in detaliu si pas cu pas functionalitatile, cerintele, dependentele si riscurile ce trebuie respectate in scopul functionarii la nivelul asteptat al sistemului. Se va avea in vedere si vor fi respectate cu strictete toate cerintele si obiectivele sistemului.

## Prezentare generală și abordări de proiectare

### Prezentare generală

Sistemul este format din partea hardware reprezentata de placutele ESP32 si ESP32-CAM, diferiti senzori, motoare dc, coolere de 12V, led-uri si receptoare IR, partea software, insemnand aplicatia android realizata cu ajutorul Android Studio si Arduino IDE, folosit pentru a coda placutele ESP. Partea de baza de date este realizata de Firebase, care permite transportul de date intre cele 2 parti.

Ngrok este folosit pentru a face port-forwarding placutei ESP32-Cam si astfel, transferul de date catre aplicatia oriunde, atat timp cat ambele au conexiune la internet.

### Presupuneri/ Constrângeri/ Riscuri

#### Presupuneri

Utilizatorul dispune de un dispozitiv mobil andoid.

Se presupune ca placutele ESP sunt alimentate corect si au conexiune la internet, iar telefonul care controleaza sistemul este de asemenea conectat la internet.

#### Constrângeri

Sistemul nu poate functiona daca telefonul sau placutele ESP nu au conexiune stabila la internet.

#### Riscuri

Poate exista riscul ca Firebase sa nu raspunda la solicitarile uneia sau ambele parti, dar sunt extrem de mici.

Riscul de a uita parola este acoperit de functionalitatea de a reseta parola si de autentificare prin biometrice.

Exista de asemenea riscul ca placutele sa nu fie alimentate corect ceea ce duce la infunctionalitatea lor.

## Considerații de proiectare

### Obiective și linii directoare (ghiduri)

Primul obiectiv este de a realiza o baza de date Firebase pentru a face legatura intre partea software si cea hardware.

Al 2-lea obiectiv este de a programa o aplicatie android care sa controleze si sa citeasca datele din firebase.

Ultimul obiectiv este de a realiza o macheta care sa simuleze casa smart, confectionata din lemn si plexiglas, unde vor fi montate componentele hardware.

### Metode de dezvoltare

Pentru realizarea proiectului, s-au utilizat următoarele metode și tehnologii:

* **Arduino IDE**: pentru programarea modulelor ESP32 și ESP32-CAM, folosind limbajul C++. S-a utilizat programarea orientată pe funcții pentru o mai bună modularizare a codului (de exemplu funcții separate pentru citirea senzorilor, controlul motoarelor, transmiterea datelor către Firebase).
* **Android Studio**: pentru dezvoltarea aplicației mobile în limbajele Java și Kotlin, folosind modelul de arhitectură MVVM (Model-View-ViewModel) pentru o separare clară între partea de business logic și interfata grafică.
* **Firebase**: ca platformă de backend pentru autentificare, stocare și sincronizare date în timp real.
* **Ngrok**: pentru realizarea tunelării porturilor în vederea accesului la fluxul video live oferit de ESP32-CAM din orice rețea de internet.

Astfel, dezvoltarea a fost orientată pe modularitate și scalabilitate, pentru a facilita întreținerea ulterioară sau extinderea sistemului.

### Strategii de arhitectură

Structura arhitecturală a proiectului este una distribuită și cuprinde următoarele componente principale:

* **ESP32 principal**: colectează date de la senzori (temperatură, umiditate, gaz, CO2, lumina ambientală) și controlează actuatoarele (motoare DC pentru poartă și ușă de garaj, ventilatoare/coolere pentru reglarea temperaturii).
* **ESP32-CAM**: asigură supravegherea video live, transmițând imagini printr-un server HTTP, accesibil prin intermediul unui tunel Ngrok.
* **Firebase**: servește ca punct central de sincronizare între ESP-uri și aplicația mobilă. Găzduiește date despre utilizatori, status-ul senzorilor, comenzile de control și fluxurile de notificare.
* **Aplicatie Android**: permite autentificare/logare, controlul complet al casei inteligente și monitorizarea în timp real a parametrilor mediului și a fluxului video.

Arhitectura a fost proiectată pentru a permite scalabilitate: este posibilă adăugarea de noi senzori sau actuatoare fără modificări majore în structura aplicației.

## Arhitectura Sistemului și Proiectarea Arhitecturii

### Vedere logică

Sistemul este format din trei noduri logice principale: Hardware ESP32, server Firebase și aplicația Android. Datele circulă bidirecțional: ESP32 trimite măsurători în Firebase, aplicația le citește și permite trimiterea comenzilor de control.

### Arhitectură hardware

 **ESP32 DevKit v1**: conectat la senzori (DHT22, MQ2, MQ135, senzor lumină) și la driverele de motoare pentru controlul mecanismelor de poartă și ușă.

 **ESP32-CAM**: dedicat transmisiei video.

 **Componente auxiliare**: motoare DC, led-uri cu receptori IR pentru detecția obstacolelor, coolere de 12V pentru ventilație controlată.

### Arhitectură software

 **Firmware ESP32**: gestionează comunicarea cu senzorii și actuatoarele, respectiv sincronizarea cu Firebase.

 **Firmware ESP32-CAM**: gestionează serverul de streaming video.

 **Aplicatie Android**: gestionare conturi (logare/inregistrare/resetare parola), vizualizare date, control porti/lumină/temperatură, stream video.

### Arhitectura informațiilor

 **Date utilizator**: email, parola (criptată), date de autentificare biometrice.

 **Date senzori**: valori numerică pentru temperatură, umiditate, gaz, CO2 și intensitatea luminii.

 **Date de control**: starea porții, a ușii, a luminilor, a coolerelor.

### Arhitectura de comunicații interne

 ESP32 și ESP32-CAM comunică independent cu Firebase și aplicatia prin Wi-Fi.

 Datele de control și cele de monitorizare sunt sincronizate prin Firebase Realtime Database.

 Fluxul video este accesat printr-un link Ngrok securizat.

## Proiectarea sistemului

### Proiectarea bazei de date

#### Obiecte de date și structuri de date rezultante

 **Utilizator**: Obiect care conține email, parola criptată și datele de autentificare biometrică (opțional).

 **Date senzor**: Obiect pentru fiecare categorie de senzor (temperatură, umiditate, lumină, gaz, CO2) ce stochează ultima valoare măsurată și timestamp-ul citirii.

 **Comenzi control**: Obiecte ce definesc acțiuni asupra porții, ușii de garaj, luminii și ventilatoarelor.

#### Fișiere și baze de date

 **Firebase Realtime Database**: folosit pentru a stoca în timp real datele senzorilor și comenzile.

 **Firebase Authentication**: gestiunea utilizatorilor (conturi, resetări de parole, autentificare securizată).

### Conversii de date

 Datele de la senzori sunt preluate în format brut și convertite în valori semnificative (ex: tensiuni citite de ADC → temperatura/umiditate/gaz).

 Fluxul video de la ESP32-CAM este prelucrat într-un format MJPEG pentru transmisie eficientă prin rețele WiFi.

 Comenzile trimise de utilizator sunt convertite în semnale logice (HIGH/LOW) pentru controlul motoarelor și ledurilor.

### Interfețe utilizator

#### Intrări

 Formular de autentificare/inregistrare/resetare parola.

 Panou de control cu butoane pentru fiecare element al casei (poarta, usa garaj, lumini, ventilatoare).

 Setări cont (schimbare date personale, activare autentificare biometrică).

#### Ieșiri

* Afișare valorilor curente de la senzori.
* Feedback vizual pentru starea elementelor controlate (deschis/închis, activat/dezactivat).
* Afișare flux live de la camera ESP32-CAM.

### Proiectarea interfețelor cu utilizatorul

A screen shot of a room

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect. A screenshot of a login screen

AI-generated content may be incorrect.

## Scenarii de utilizare

Scenariu 1: Autentificare

Utilizatorul deschide aplicația.

Introduce email și parolă sau folosește autentificarea biometrică.

Dacă datele sunt corecte, utilizatorul este autentificat și accesează dashboard-ul.

Scenariu 2: Monitorizare mediului

Utilizatorul accesează pagina de senzori.

Sunt afișate valorile actualizate pentru temperatură, umiditate, nivel gaz, CO2 și luminozitate.

Scenariu 3: Control poartă/garaj

Utilizatorul apasă butonul „Deschide poarta” sau „Închide poarta”.

Comanda este transmisă prin Firebase către ESP32.

Motorul DC pornește/oprește corespunzător, iar starea este actualizată în aplicație.

Scenariu 4: Vizualizare flux video

Utilizatorul accesează pagina „Camera”.

Aplicația deschide stream-ul video live de la ESP32-CAM prin link-ul Ngrok.

Scenariu 5: Notificare obstacol

În timpul închiderii porții, un obstacol este detectat de senzorul IR.

ESP32 opreste inchiderea portii si o deschide pentru a evita accidentele.

## Proiectare de detaliu

### Proiectare hardware de detaliu

• ESP32 conectat la:

• DHT22 (temperatură și umiditate) - pin digital.

• MQ2 și MQ135 (gaz, CO2) - pinii analogici.

• Senzor lumină - pin analogic.

• LED IR + receptor IR pentru detectare obstacole - pinii digitali.

• Driver motoare (L298N sau similar) pentru comanda motoarelor DC.

• Coolere 12V alimentate separat cu comandă via releu.

• ESP32-CAM:

• Conexiune directă la sursa de alimentare 5V.

• Server HTTP configurat pentru streaming video.

### Proiectare software de detaliu

 Firmware ESP32:

* Interfață de conectare WiFi.
* Interfață Firebase pentru trimitere/citire date.
* Funcții dedicate pentru fiecare senzor și fiecare actuator.

 Firmware ESP32-CAM:

* Server HTTP pentru flux video.
* Updatarea periodică a URL-ului Ngrok.

 Aplicație Android:

* Activități și fragmente pentru autentificare, dashboard senzori, control elemente, vizualizare camera.
* Utilizarea LiveData și ViewModel pentru gestionarea fluxurilor de date.

### Proiectare detaliată de securitate

 Criptarea comunicării între aplicație și Firebase folosind TLS/SSL.

 Hash-uirea parolelor utilizatorilor.

 Resetare parolă cu email de verificare.

 Acces la date protejat cu reguli de securitate Firebase.

### Proiectare de detaliu pentru performanța sistemului

 Limitarea frecvenței de update a datelor la fiecare 5 secunde pentru reducerea traficului.

 Stream video MJPEG optimizat pentru minimizarea delay-ului.

 Optimizare consum ESP32 prin folosirea modurilor de economisire a energiei.

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

 MQTT/HTTP pentru comunicarea între aplicație și hardware.

 Actualizare date senzor la fiecare citire nouă.

 Comenzi control sincronizate imediat în baza de date pentru răspuns rapid.

## Controale pentru verificarea integrității sistemului

 Logarea tuturor acțiunilor utilizatorilor în Firebase (audit trail).

 Validarea valorilor de intrare (de exemplu temperaturi între -40\u00baC și 125\u00baC pentru DHT22).

 Backup zilnic al bazei de date Firebase.

 Detectarea automată a deconectării modulelor ESP și notificarea utilizatorului.

 Restricționarea accesului în aplicație pentru utilizatori neautentificați.

Anexa A: Gestiunea modificărilor documentului

Tabel 1 – Înregistrarea modificărilor asupreaa documentului curent

| versiune | Data | Autorul/Deținătorul | Descriere |
| --- | --- | --- | --- |
| 1.0 | 28/04/2025 | Valentin Lazar | Completare date |