

SMART HOME AUTOMATION HUB

ANDREI Simona, GHEORGHE Razvan, GHINEA Bogdan, NEAGU Alexandru

Conducător științific: Dr.ing. ALUPEI-COJOCARIU OVIDIU-DORIN

REZUMAT: Conceptul de “smart home” sau de “smart house” este unul care se dezvoltă pe o scară din ce în ce mai largă și care definește un ambient (casa, clădire, companie, etc) care încorporează sisteme inovatoare și automatizate ce oferă celor care locuiesc sau își desfășoară activitatea în acel ambient o modalitate de utilizare, control și monitorizare a funcționalităților cu care este dotată respectiva clădire.

Funcția generală a produsului Smart Home Automation Hub poate fi descrisă în felul următor: capacitatea de gestionare într-un mod programabil a echipamentelor electronice din domiciliu. Aceasta succintă enunțare este în strânsă legătură cu definiția conceptului home-automation, cunoscut în România și sub numele domotica : extensia firească a automatizării clădirilor rezidențiale ce implică controlul iluminării, încălzirii, ventilației sau echipamentelor de refrigeratie, pe lângă electrocasnice precum frigider, mașini de spălat, cafetiere sau orice alt tip de echipament compatibil.

1 INTRODUCERE

În cadrul acestei lucrări se va prezenta proiectarea detaliată a produsului Smart Home Automation Hub, cât și funcțiile pe care acesta le are.

Funcția generală a produsului Smart Home Automation Hub poate fi descrisă în felul următor : capacitatea de gestionare într-un mod programabil a echipamentelor electronice din domiciliu. Aceasta succintă enunțare este în strânsă legătură cu definiția conceptului home-automation, cunoscut în România și sub numele domotica : extensia firească a automatizării clădirilor rezidențiale ce implică controlul iluminării, încălzirii, ventilației sau echipamentelor de refrigeratie, pe lângă electrocasnice precum frigider, mașini de spălat, cafetiere sau orice alt tip de echipament compatibil.

2 PROIECTAREA DETALIATĂ

Produsul nostru se compune dintr-o componentă software și una hardware, ce împreună realizează produsul Smart Home Automation Hub. În cadrul acestui capitol vom prezenta proiectarea detaliată pentru componenta software din punct de vedere al protocoalelor și a tehnologiei folosite.

2.1 Arhitectura software generală

Pornind de la funcțiile principale ale produsului, se pot observa în figura 1 blocurile principale ce alcătuiesc componenta software a produsului. Aceste blocuri poartă nume descriptive, intitulate în funcția de principală funcție și sunt organizate în două categorii generale, după următorul criteriu

: dacă utilizatorul final are acces la respectivele resurse, sau nu. Astfel putem enumera cele două grupe de funcții : funcțiile din frontend și funcțiile din backend.

În dezvoltarea software, cei doi termeni separă nivelul logic de prezentare a datelor și nivelul de acces al datelor, ori infrastructura hardware. Ca exemplu se poate considera modelul client-server : în această ecuație, clientul este considerat parte din frontend, iar serverul este considerat backend. În plus, se poate observa o diferență fundamentală prin scopul acestora : în nivelul de prezentare, datele sunt prezentate în urma operațiilor de stocare/analiză realizate în backend. O altă regulă definitorie spune că frontendul este în proximitatea clientului, în timp ce backendul se poate afla la mare distanță, într-o locație necunoscută utilizatorului final.

Echipamentul de automatizare a locuinței descris în această lucrare, Smart, include următoarele funcții:

Frontend

- Interfața Administrator : interfața accesibilă unui număr restrâns de utilizatori (standard 1 utilizator sub alias-ul *Admin*) cu drepturi extinse de administrare și depanare; Aceasta interfața permite administrarea hub-ului, dar impune o utilizare precaută deoarece o utilizare improprie poate duce la o defecțiune.
- Interfața Utilizator : interfața simplificată, prin intermediul căreia se pot citi date de monitorizare și manipula elemente

dinamice gestionate de acest echipament; datele furnizate nu beneficiaza de o granularitate la fel de mare ca in cazul interfetei de administrare.

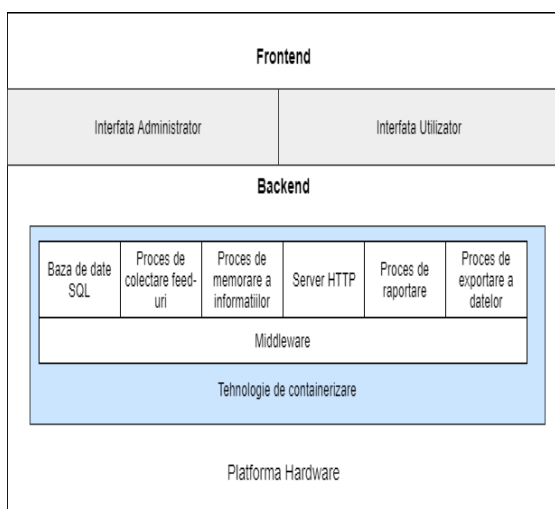


Fig. 1 Arhitectura functionala software

2.2 Blocuri functionale pentru FrontEnd

In proiectarea si implementarea componentelor din frontend au fost considerate urmatoarele principii comune :

- Ergonomicitate, interfata simpla, intuitiva, usor de utilizat
- Performante optime : timpi redus de incarcare, utilizarea scazuta a resurselor de procesare
- Accesibilitate : posibilitatea de a fi utilizat de pe orice dispozitiv mobil, fara a fi nevoie de a instala un utilitar dedicate
- Securitate : protectia datelor utilizatorului, restrictia accesului nepermis

Cele doua blocuri functionate din frontend au fost proiectate pentru a fi accesate cu ajutorul unui browser web. Limbajul de programare utilizat, dependetele si modul in care aceste programme au fost scrise fac posibila utilizarea oricaror navigatoare web, dintre care enumeram compatibilitatea cu Internet Explorer, Chrome, Firefox si Safari.

Accesarea blocului dorit se face prin intermediul unui portal de autentificare, disponibil fie prin intermediul unei adrese IP fie printr-un FQDN (fully qualified domain name). Utilizarea uneia din cele doua variante impune anumite avantaje/dezavantaje :

Adresa IP : poate fi accesibila din reseaua locala (de exemplu reseaua wireless din imobil) fara a necesita alte setari/modificari in infrastructura; fie

nu este nevoie de configurare, fie setarea adresei IP este extrem de facila;

Dezavantajul este generat de faptul ca pentru accesul remote (din afara retelei locale) anumite configurari trebuiesc realizate pe echipamentul Layer 3 din reseaua locala; poate reprezenta o potentiala bresa de securitate;

FQDN : accesul se poate realiza atat din reseaua locala, cat si din cea externa; este usor de memorat, poate fi personalizata (ex www.cercetarestiintifica.edu) ; Dezavantajul il reprezinta dificultatea de instalare/configurare si eventuale costuri pentru domenii ce trebuiesc achizitionate;

Permisuniunea de acces este verificata in urma unei autentificari prin doua mijloace : utilizator si parola. In urma introducerii credentialelor, accesul este permis in cel putin una din functiile din frontend (Interfata Utilizator sau Interfata Administrator) pe baza nivelului de acces asociat clientului. In figura 2, se poate observa ca in urma autentificarii cu datele de acces pentru contul de administrator, exista doua cai de acces catre cele doua interfete de utilizare.

SMARTYR

Please login with your credentials.

Username

Password

Login

SMARTYR

Logged in as Admin
Go to [\[Admin Dashboard\]](#) [\[User Dashboard\]](#) or [\[Logout\]](#)

Fig. 2 Portal de Autentificare in Frontend

Cele doua valori introduse in campurile de autentificare sunt comparate cu datele de autentificare asociate contului, stocate in baza de date. In cazul in care aceste date nu corespund, utilizatorul va fi alertat grafic (vezi figura 3) iar evenimentul va fi memorat in log-urile aplicatiei.

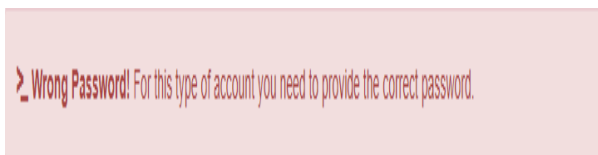


Fig 3. Notificarea respingerii accesului

2.3 Interfata Administrator

Interfata de administrare permite o buna granularitate in raportarea valorilor masurate si efectuarea unor operatiuni de gestionare a functionarii echipamentului, intr-o maniera minimalista si intuitiva. La momentul redactarii acestei lucrari, din interfata de administrare se pot realiza urmatoarele :

- Listarea tuturor valorilor primite de la senzori si reprezentarea grafica a acestora
- Filtrarea valorilor receptionate pe diferite criterii : locatie, tipul parametrului si numarul de intrari
- Calculul rapid al umiditatii si temperaturii medii pentru valorile listate
- Backup-ul bazei de date si descarcarea acestora
- Prezentarea unei statistici pentru umiditate defalcate pe incaperi
- Prezentarea unei statistici pentru temperatura defalcate pe incaperi

Pentru dezvoltari ulterioare, sunt prevazute urmatoarele functionalitati ce vor fi introduce in interfata de administrare :

- Adaugarea/eliminarea/repornirea senzorilor
- Export-ul logurilor aplicatiei (pentru depanare)
- Crearea unui panou de afisare a evenimentelor

Buna functionare a echipamentului este notificata inca de la accesul in interfata de administratie, fapt notificat in partea superioara a interfetei web prin mesajul din figura 4. Finalizarea cu succes a operatiei de deschidere a bazei de date garanteaza o buna functionare a tuturor celorlalte elemente ale solutiei software (middleware, librarii, fisiere etc.)

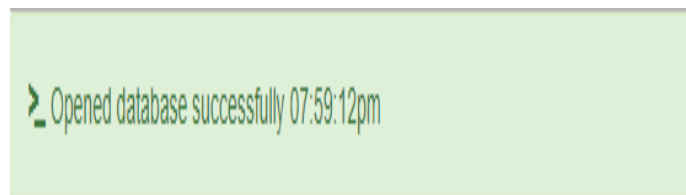


Fig. 4 Notificarea bunei functionari a solutiei

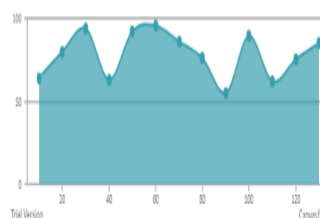


Logged in as : Admin

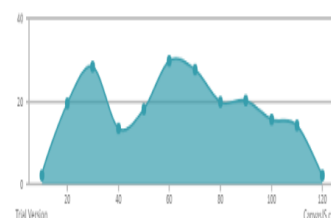
[Dashboard] [Average Humidity] [Average Temperature]

Choose here the location Choose here the parameter 25

Humidity



Temperature



Sensor Name	Sensor Type	Position	Value Type	Value	Date
Dummy-2	Temperature + Humidity	Room-1	Humidity	63.69	29-Apr-2017 16:52:19:507051
Dummy-1	Temperature + Humidity	First Room	Temperature	1.97	29-Apr-2017 16:52:16:505001
Dummy-2	Temperature + Humidity	Room-1	Humidity	79.99	29-Apr-2017 16:52:13:502806
Dummy-2	Temperature + Humidity	Room-1	Temperature	19.41	29-Apr-2017 16:52:10:500707
Dummy-3	Temperature	First Room	Humidity	93.39	29-Apr-2017 16:52:07:496259
Dummy-1	Temperature + Humidity	First Room	Temperature	28.25	29-Apr-2017 16:52:04:497891
Dummy-4	Humidity	Living Room	Humidity	63.22	29-Apr-2017 16:52:01:496533
Dummy-2	Temperature + Humidity	Room-1	Temperature	13.3	29-Apr-2017 16:51:58:494220
Dummy-4	Humidity	Living Room	Humidity	92.25	29-Apr-2017 16:51:55:492909

Average Humidity

78.19 %

Average Temperature

17.46 C

Backups

backup-2017-04-29-20:40:59.bak
 backup-2017-04-29-20:41:45.bak
 backup-2017-04-29-20:44:55.bak
 backup-2017-04-29-15:55:51.bak

Fig. 5 Dashboard-ul interfetei de administrare

Dashboard-ul interfetei de administrare reprezinta cel mai important panou de utilizare. Prin intermediul acestuia, se pot lista valorile receptioante de la senzori, filtrarea acestora si reprezentarea grafica pentru studiul evolutiei in timp.

In partea superioara se pot observa filtrele de raportare. In functie de valorile returnate in urma acestei interogari, cele doua grafice sunt trasate

conform. Aproximarea ochiosmetrica poate fi dublata de calculul valoriloee medii, disponibile in partea dreapta a panoului atat pentru umiditate cat si pentru temperatura.

Pentru crearea unui backup manual, se poate utiliza butonul disponibil in partea drapta a interfetei. Sub acesta, sunt listate backup-urile existente, ce pot fi downloadate cu scopul de a fi salvate pe termen lung.

In meniul *Average Temperature* se pot observa valorile cele mai recente ale temperaturii pentru fiecare incinta in parte. In figura 6 se poate observa faptul ca aceste valori nu coincide cu datele receptionate de la senzori, ci sunt valorile calculate cu ajutorul unei medii mobile, pentru o mai mare acuratete.

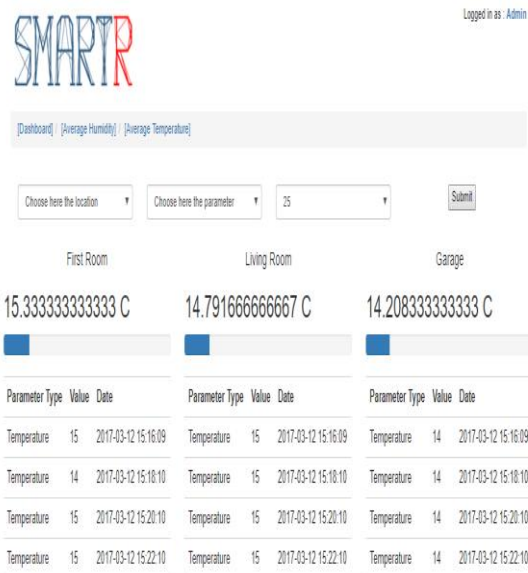


Fig. 6 Panoul *Average Temperature* pentru toate incintele

2.4 Interfata Utilizator

Interfata utilizatorului este cea care permite utilizarea uzuala a hubului Smartr, motiv pentru care design-ul este extrem de simplu. Aceasta abordare permite respectarea tuturor principiilor enuntate in inceputul lucrarii.

SMARTR

Logged in as Admin
Connected from 192.168.31.1



Fig. 7 Planul casei, reprezentare 3D Utilizarea este extrem de intuitiva : pentru raportarea valorilor masurate intr-o anumita incinta, si controlul elementelor active din incapere este suficient sa accesezi cu ajutorul unui click pe una din camerele din imobil. Astfel, o fereastră aferenta camerei va fi deschisa, fereastră cu ajutorul careia se pot efectua actiunile disponibile.

Cu scop demonstrativ, prototipul echipamentului permite monitorizarea datelor de temperature si umiditate, dar si controlul luminilor din incinta. Este important de inteles conceptul : orice parametru sau element active se poate implementa cu usurinta, aplicatia fiind gandita sa faciliteze astfel de implementari.

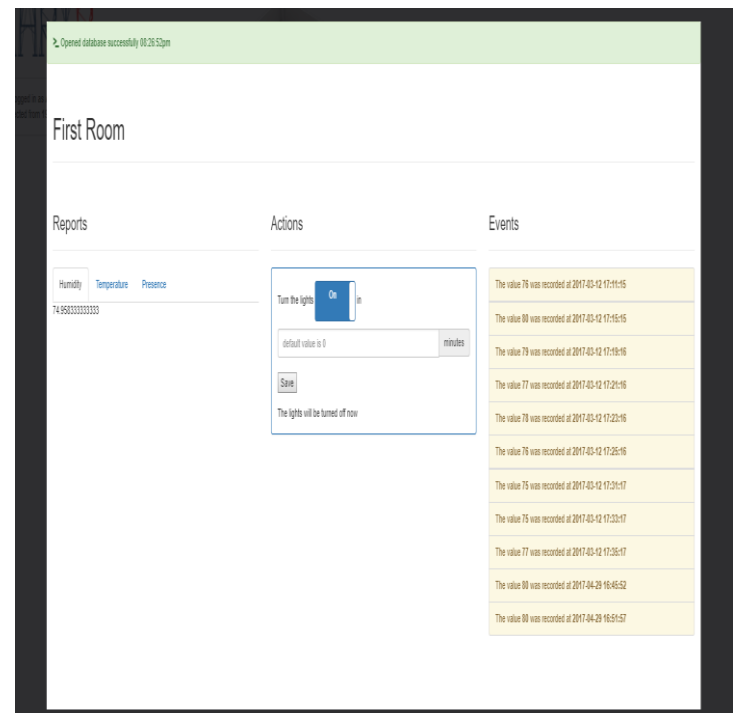


Fig. 8 Panoul detaliat al camerelor

2.5 Blocuri fundamentale pentru BackEnd

Funcțiile principale din backend au fost decuplate, fiind îndeplinite de programe/scripturi interconectate; motivul alegerii acestei abordări modulare îl reprezintă posibilitatea de a îmbunătăți/modifica module specifice fără a fi afectată buna funcționare per total. În plus, adăugarea unor noi module, depanarea și optimizarea soluției este mult simplificată. Funcțiile din backend sunt îndeplinite de coduri și pachete software dedicate, acestea putând fi descrise după funcția principală după cum urmează :

- Baza de date SQL : folosită pentru a stoca datele ingerate pe termen scurt și a datelor sumarizate, pe termen lung;
- Proces de colectare feed-uri : fragmentul de cod responsabil cu recepționarea datelor, prin diferite protocoale (MQTT/AMQP); Prin feed de date înțelegem un mecanism de primire a datelor de la sursă, utilizat în mod normal în aplicații real-time dar și în World Wide Web;
- Proces de memorare a informațiilor : porțiunea de cod responsabilă cu stocarea datelor preluate de la procesul de colectare a feed-urilor și stocarea lor într-un format standardizat, în baza de date;
- Server HTTP : program menit să răspundă cererilor de conținut provenite din browsere web, pentru încărcarea conținutului din frontend;
- Proces de raportare : responsabil cu generarea interfețelor din frontend.
- Proces de exportare a datelor : fragmentul de cod responsabil cu crearea copiilor de siguranță, disponibile pentru a fi descărcate din frontend;

2.6 Interconectarea blocurilor functionale pentru backend

În figura 9 sunt reprezentate interconectările între aceste blocuri functionale, și schimbul de informație dintre acestea, prin săgeți uni sau bi-direcționale. Săgețile bi-direcționale reprezintă faptul că blocurile functionale pot avea atât rolul de sursă cât și destinație.

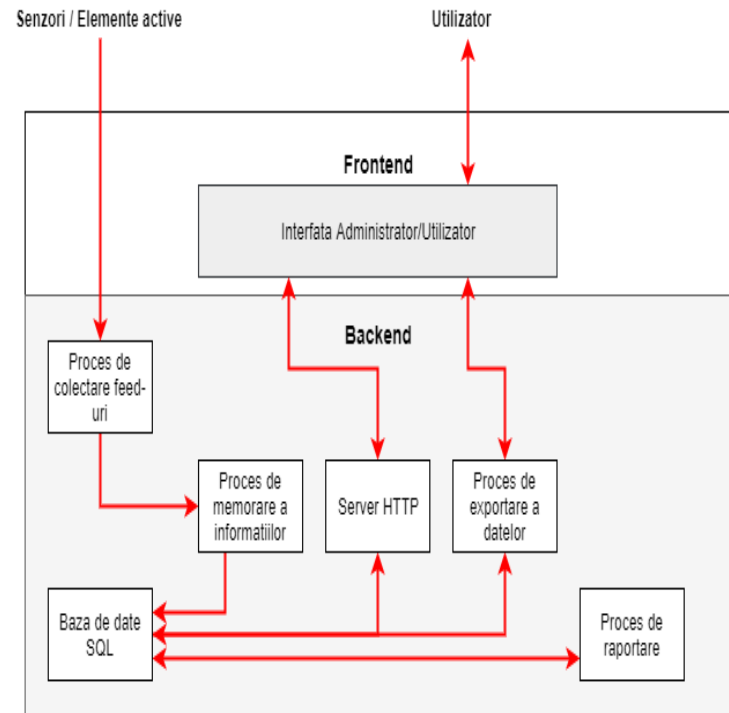


Fig. 9 Interconectarea blocurilor functionale backend

2.7 Baza de date SQL

Pentru stocarea informațiilor colectate atât pe termen scurt, cât și pe termen lung s-au analizat multiple soluții de baze de date relaționale. Motivul alegerii unei astfel de soluții îl reprezintă performanța sportită oferită de limbajul SQL : se pot stoca într-un model structurat, cantități mari de date într-o manieră simplificată. Bazele de date SQL utilizează standarde maturizate, adoptate de organisme precum ANSI & ISO iar programarea acestor structuri de date este facilă, printr-un limbaj intuitiv.

Caracterul dinamic al acestor soluții face posibilă atât modificarea online a datelor stocate, dar și modificarea structurilor, tabelelor și a relațiilor dintre acestea în timp real. În plus, există multiple soluții tehnologice disponibile pentru implementare, dezvoltate de companii de renume din industrie.

SQL poate fi considerată una din principalele tehnologii ce propulsează avântul tehnologic în industria web&mobile.

Soluțiile de stocare a informației relaționale pleacă de la premiza că există un model de salvare a datelor, așa numită "schema de date". Aceste scheme de date sunt guvernate de relații, iar relațiile la rândul lor sunt implementate cu ajutorul unui sistem de management (RDMS). Fiecare intrare în baza de date se poate traduce într-un rând, unic identificabil, stocat în cadrul unui tabel.

Deși principiul rămâne același, există diferite implementări ale aceleși logici SQL, fiecare cu avantaje și dezavantaje. Diferențele diferitelor soluții pot fi de tipul datelor ce pot fi salvate, a operațiilor aplicate respectivelor date, scalabilitate, consum de resurse etc. Pentru alegerea tehnologiei SQL ce este folosită în produsul descris în prezenta lucrare, au fost analizate soluțiile majore open-source disponibile, printre care :

- MySQL
- Postgres
- SQLite

MySQL

Este cea mai populară soluție destinată mediilor mari, scalabilă bazată pe arhitectura tip server. Include multe funcționalități fiind un produs open-source ce propulsează majoritatea aplicațiilor online. Datorită acestei popularități, există foarte multe soluții software 3rd party ce permit interoperabilitatea cu MySQL.

Chiar dacă nu oferă întregul standard SQL, MySQL oferă multe din funcționalități utilizatorilor. Fiind o bază de date separată, aplicațiile discută cu daemon-ul MySQL pentru a accesa baza de date propriu-zisă.

Avantajele MySQL

- Usor de lucrat: MySQL poate fi instalat foarte rapid. Aplicațiile third party pot comunica foarte ușor cu această bază de date.
- Funcționalități multe: MySQL oferă multe din funcționalitățile SQL care sunt așteptate de la o bază de date (direct sau indirect).
- Securitate: multe capabilități are incorporate în MySQL.
- Scalabil și puternic: MySQL poate să manipuleze cantități mari de date și poate scala în funcție de dimensiunea load-ului.
- Viteza de execuție: Renunțarea la unele protocoale standard oferă MySQL opțiunea de a lucra mult mai rapid și mai eficient

Dezavantajele MySQL

- Limitări cunoscute:

Prin design, MySQL nu e conceput să aibă toate capabilitățile de care aplicațiile aparute recent au nevoie.

- Probleme de fiabilitate:

Modul în care unele funcționalități sunt suportate de MySQL o fac mai puțin fiabile în comparație cu alte baze de date.

- Dezvoltare stagnată

Chiar dacă MySQL este un produs open source, în ultimul timp au apărut plângeri cu privire la procesul de dezvoltare al acesteia. Cu toate acestea trebuie să avem în vedere că sunt baze de date care pot fi adăugate peste instalările standard de MySQL (ex: MariaDB).

Când să folosim MySQL

- Operațiuni distribuite

Când avem nevoie de mai multe capabilități decât ce oferă SQLite, MySQL poate aduce mai multe posibilități avansate.

- Securitate ridicată

MySQL are capabilități de securitate pentru protecția datelor și a accesului la date.

- Web-site-uri și aplicații web

Majoritatea aplicațiilor web funcționează cu ajutorul MySQL pentru că este o bază de date flexibilă și ușor de administrat.

- Soluții personalizate

Dacă sunt folosite aplicații personalizate, MySQL poate interacționa foarte ușor cu acestea datorită plajei bogate de configurări și de modalități de operare.

Când să nu folosim MySQL

- Scrieri concurente:

Chiar dacă citirile concurente nu sunt o problemă pentru MySQL, scrierile concurente sunt problematice.

- Lipsa unor capabilități

MySQL poate să nu aibă anumite capabilități necesare pentru activitățile de zi cu zi.

PostgreSQL

PostgreSQL este un sistem de baze de date relational. Este disponibil gratuit sub o licență open-source de tip BSD. PostgreSQL nu este controlat de nici o companie, își bazează dezvoltarea pe o comunitate răspândită la nivel global, precum și câteva companii dezvoltatoare. Acesta încearcă să adopte standardele ANSI/ISO SQL la fiecare nouă revizie software.

Comparabil cu alte sisteme de baze de date relationale, PostgreSQL diferă prin suportarea a multor capabilități și funcționalități necesare în comunicarea cu alte aplicații și baze de date.

Datorita tehnologiei din care este realizata, PostgreSQL este capabil sa suporte mai multe taskuri concomitand. Acest lucru este realizat in principal datorita Multiversion Concurrency Control (MVCC).

PostgreSQL este un sistem programabil, cu procedure personalizabile ce sunt numite "stored procedures". Aceste functii pot fi create pentru a simplifica executia unor activitati ce sunt repetabile.

Avantajele PostgreSQL

- Un sistem de baze de date relational opensource cu SQL standard:

PostgreSQL este un sistem opensource gratuit, insa foarte puternic.

- Comunitate puternica:

PostgreSQL beneficiaza de o comunitate devotata si experimentata ce presupune si un knowledge-base foarte puternic, ce poate fi accesat 24/7 gratuit.

- Suporta aplicatii third-party:

PostgreSQL beneficiaza de un numar foarte mare de aplicatii third party suportate.

Dezavantajele PostgreSQL

- Performanta:

Pentru un volum mare de citiri simple, PostgreSQL poate parea mai putin performant decat MySQL.

- Popularitate:

Comunitatea pentru MySQL este mult mai mare decat cea a lui PostgreSQL, fapt ce poate afecta posibilitatea de a primi suport pentru diferitele activitati.

- Hosting:

Datorita factorilor mentionati mai sus, este din ce in ce mai greu sa gasim prestatori de servicii ce ofera PostgreSQL ca si solutie.

Cand sa folosim PostgreSQL

- Integritatea datelor

Cand fiabilitatea si integritatea datelor sunt foarte importante, PostgreSQL este o solutie buna.

- Solutii personalizate:

Daca baza de date are nevoie de procedure personalizate, PostgreSQL este o solutie buna.

- Integrabilitatea:

Daca se doreste migrarea bazei de date la o solutie proprietara (ex: Oracle), PostgreSQL poate face acest lucru mult mai usor datorita protocoalelor standard.

- Design complex:

Comparativ cu alte solutii opensource, pentru bazele de date complexe PostgreSQL ofera majoritatea functionalitatilor necesare.

Cand sa nu folosim PostgreSQL

- Viteza:

Daca sunt necesare doar citiri foarte rapide, atunci PostgreSQL nu este aplicatia perfecta.

- Configurari simple:

Daca sunt necesare doar configurari simple, atunci PostgreSQL poate fi mult prea complex pentru asta.

SQLite

SQLite este o mică bibliotecă C care implementează un motor de baze de date SQL încapsulat, oferă posibilitatea de a-l introduce în diverse sisteme și necesită zero-configurare. Când o aplicatie foloseste SQLite, integrarea functioneaza direct cu baza de date SQLite, fata de o comunicare prin interfete si porturi. Acest lucru face ca SQLite sa fie foarte rapid si puternic.

Avantajele SQLite

- Bazat pe fisier:

Baza de date este salvata pe un singur fisier, ceea ce o face foarte portabila.

- Foloseste Standardele SQL:

Chiar daca beneficiem de o baza de date simplificata, SQLite dispune de capabilitatile standard pentru SQL-uri (JOIN,SELECT,etc).

- Foarte buna pentru dezvoltare si testare:

In timpul dezvoltarii aplicatiilor, pentru majoritatea dintre acestea este necesara o solutie ce suporta o concurenta mare de citiri si scrieri. De aceea SQLite este o solutie foarte buna pentru dezvoltare, datorita si simplitatii acesteia.

Dezavantajele SQLite

- Nu poate creste in performanta

Datorita faptului ca este o realizare simpla din punct de vedere tehnic, biblioteca ce sta la baza SQLite, baza de date nu poate fi crescuta in performanta decat este deja.

Cand sa folosim SQLite

- Aplicatii incorporate:

Toate aplicatiile ce au nevoie de portabilitate, ce nu au nevoie de crestere, cum ar fi aplicatiile mobile sau jocurile.

- Testare

Faptul ca toata baza de date este compusa dintr-un singur fisier face ca testarea sa cu ajutorul SQLite fie foarte simpla si usor de realizat .

3 CONCLUZII

Avand in vedere cele enuntate mai sus, speram ca prin produsul nostru sa devenim un competitor pentru produsele aflate acum pe piata din Romania. Dorim ca in continuare sa ne axam pe definitivarea partii hardware pentru un design cat mai compact si cat mai calitativ ce va putea fi achizitionat de catre viitorii nostrii client la un pret mult mai redus fata de actuala competitive prezenta pe piata din Romania.

4 BIBLIOGRAFIE

[1].Smith,George disponibil la <https://www.tutorialspoint.com/> Accesat la data: 10.05.2017

[2].Griffin,Andrew disponibil la <https://www.tutorialspoint.com/mysql/> Accesat la data: 10.05.2017.

[3] Niettmaier,David disponibil la <https://www.tutorialspoint.com/postgresql/> Accesat la data : 10.05.2017

[4] Wallace, John disponibil la <https://www.tutorialspoint.com/sqlite/> Accesat la data :10.05.2017