



Documentul de proiectare arhitecturală

Realizatori

Grigore Lucian-Florin

Olteanu Eduard-Florin

Tanasă Ioan

Vasilache Andra-Gabriela

APR 15, 2021

336CA

UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI

Facultatea de Automatică și Calculatoare

1. Introducere	2
1.1 Scopul sistemului	2
1.2 Definiții, acronime	3
1.3 Documente referințe	4
2. Obiective de proiectare	5
3. Arhitectura propusă	6
3.1 Prezentare generală a arhitecturii sistemului	6
3.2 Descompunerea în subsisteme și responsabilitate fiecărui subsistem	8
3.3 Distribuția subsistemelor pe platforme hardware/software (diagrama de distribuție)	9
3.4 Managementul datelor persistente	10
3.5 Controlul accesului utilizatorilor la sistem	11
3.6 Fluxul global al controlului	12
3.7 Condițiile limită (cazurile de utilizare limită)	13

1. Introducere

1.1 Scopul sistemului

HotelHub este o aplicație care vine atât în sprijinul managerilor, cât și în sprijinul clienților unui lanț hotelier care vor să închirieze, managerieze și să documenteze (sau să se documenteze) asupra diverselor facilități ale locațiilor disponibile.

Aplicația va permite angajaților hotelului să vadă și să administreze rezervările clienților, să adauge camere și locații, dar și să modifice facilitățile oferite de un anumit tip de camera sau de către un anumit hotel. Utilizatorii neprivilegiați ai aplicației (clienții) pot alege în care dintre locațiile lanțului hotelier doresc să meargă și să facă o rezervare, pot vedea alte facilități oferite de hotelul ales și pot verifica disponibilitatea unui tip de cameră pentru o perioadă selectată.

1.2 Definiții, acronime

- ❖ **Frontend** - serviciu ce livrează și este responsabil pentru interfața utilizatorului.
- ❖ **Backend** - serviciu ce este responsabil de logica dintre utilizatori.
- ❖ **DB (Bază de date)** - modalitate de a stoca informații și date pe un suport extern, cu posibilitatea extinderii ușoare și a regăsirii rapide a acestora.
- ❖ **SQL (Structured Query Language)** - limbaj specific domeniului utilizat în programare și conceput pentru gestionarea datelor.
- ❖ **MySQL Workbench** - instrument vizual unificat pentru arhitecții de baze de date, dezvoltatori și DBA.
- ❖ **IntelliJ IDEA** - mediu de dezvoltare integrat scris în Java pentru dezvoltarea de software de calculator.
- ❖ **Spring Framework** - platforma open-source care facilitează scrierea de cod în Java.
- ❖ **CSS (Cascading Style Sheet)** - limbaj de stilizare al elementelor HTML.

- ❖ **HTML (HyperText Markup Language)** - formă de marcare orientată către prezentarea documentelor text pe o singură pagină, utilizând un software de redare specializat, numit agent utilizator HTML, cel mai bun exemplu de astfel de software fiind browserul web.
- ❖ **React Native** - framework de aplicații mobile open-source creat de Facebook.
- ❖ **Javascript** - Limbaj de Programare Orientat Obiect folosit pentru logica sistemului și pentru interacțiunea elementelor din paginile Web
- ❖ **MVC (Model View Controller)** - concept foarte răspândit în programarea Web - scopul MVC este de a ține separate logica business-ului și interfața utilizator.
- ❖ **SDK (Software Development Kit)** - colecție de instrumente de dezvoltare software într-un singur pachet instalabil.

1.3 Documente referințe

[Documentul de specificație al cerințelor](#)

[Planificarea implementării](#)

2. Obiective de proiectare

Calitatea produsului software va fi asigurată de cerințele bine specificate, aplicația fiind intuitivă și ușor de folosit, precum și facil de menținut. Programul va fi capabil să satisfacă principalele cerințe funcționale: creare cont, autentificare, rezervare cameră, accesare facilități.

1. **Îndeplinirea cerințelor de performanță:** Capacitatea aplicației de a gestiona numărul de utilizatori estimat și de a răspunde într-un timp destul de rapid cererilor acestora. Serverul va fi capabil să primească și să răspundă simultan la numărul estimat de utilizatori.
2. **Îndeplinirea cerințelor de calitate:** Asigurarea aspectului întregii aplicații. Aplicația va avea un frontend plăcut vizual și intuitiv de folosit.
3. **Îndeplinirea cerințelor de încredere:**
 - a. **Fiabilitatea** - atât timp cât conexiunea la internet este asigurată, programul va garanta buna funcționare.
 - b. **Disponibilitatea** - aplicația va fi disponibilă în permanență
 - c. **Toleranța la defecte** - erorile cu care s-ar putea confrunta aplicația sunt cele cauzate de rețea. Nu sunt prevăzute erori cauzate de software-ul aplicației.
4. **Îndeplinirea cerințelor de mentenanță:** Aplicația va putea fi folosită atât de pe mobile cât și de pe web, se va adapta continuu la cerințele pieței și ale clienților. Toți cei care se vor ocupa de implementarea unor noi funcții vor trebui să rezolve și anumite probleme care vor apărea pe parcurs după punerea în funcțiune a aplicației.
5. **Îndeplinirea cerințelor de planificare a proiectului:** Termenele preconizate ale realizării proiectului vor fi respectate, costul va fi unul accesibil tuturor claselor de clienți și produsele livrate vor respecta normele de calitate dorite.

3. Arhitectura propusă

3.1 Prezentare generală a arhitecturii sistemului

Avem ca scop separarea interfeței cu utilizatorul de datele și logica aplicației.

Drept stil arhitectural am ales: **Arhitectură ierarhică închisă, în 3 niveluri.**

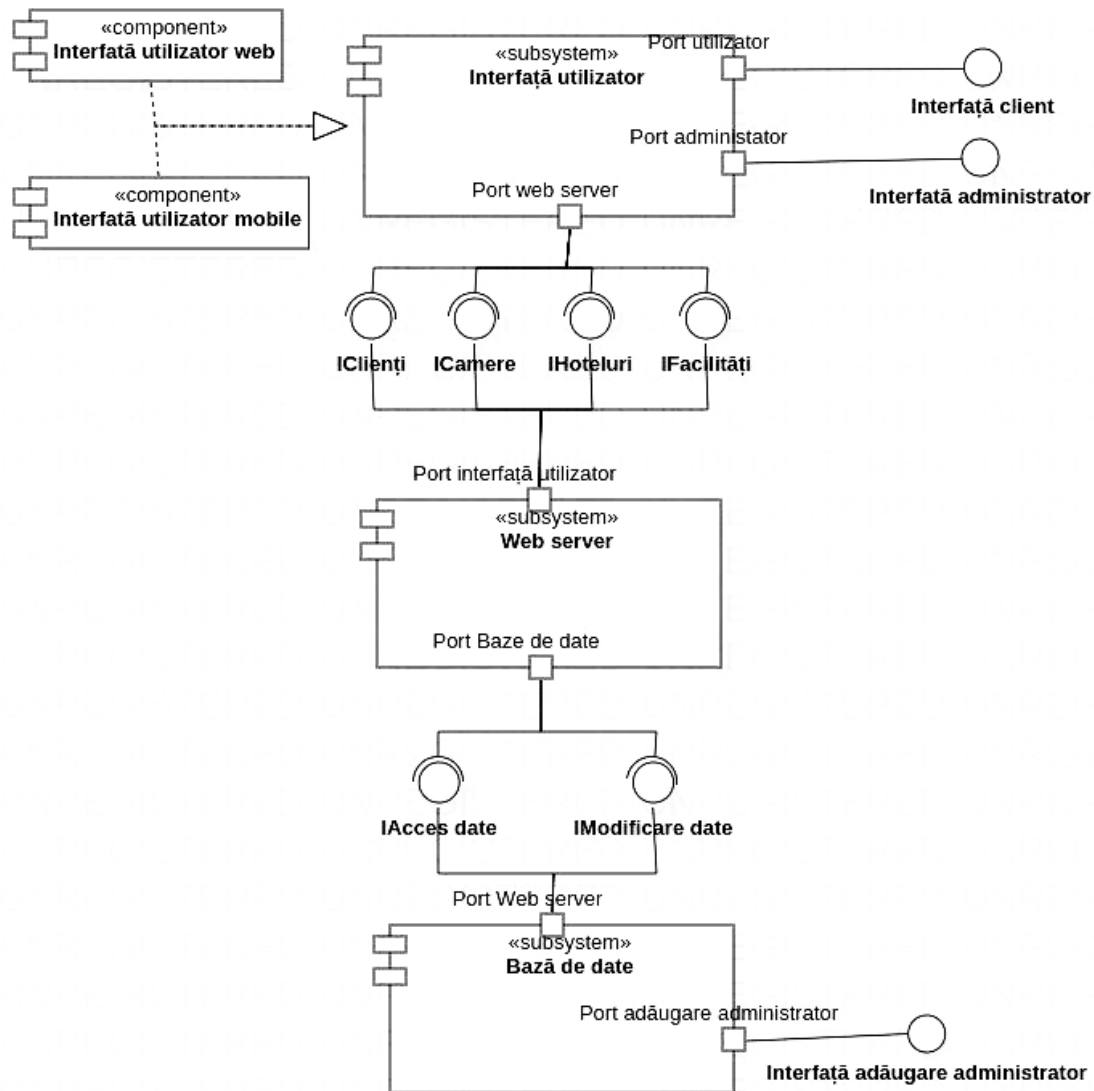
Utilizatorii aplicației pot realiza următoarele acțiuni, după cum urmează:

- **Utilizatorii neprivilegiați (conturile de clienți) :**
 - pot alege în care dintre locațiile lanțului hotelier doresc să meargă să facă o rezervare
 - pot vedea detalii despre diferitele tipuri de camere disponibile
 - pot vedea alte facilități oferite de hotelul ales
 - pot verifica disponibilitatea unui tip de cameră pentru o perioadă selectată
 - pot face o rezervare și primi confirmări din partea hotelului
- **Utilizatorii privilegiați (conturile de administratori):**
 - pot vedea rezervările clienților și le pot aproba/refuza
 - vor putea adăuga/elimina camere ale lanțului hotelier
 - de asemenea, vor putea modifica facilitățile oferite de un anumit tip de camera sau de către un anumit hotel

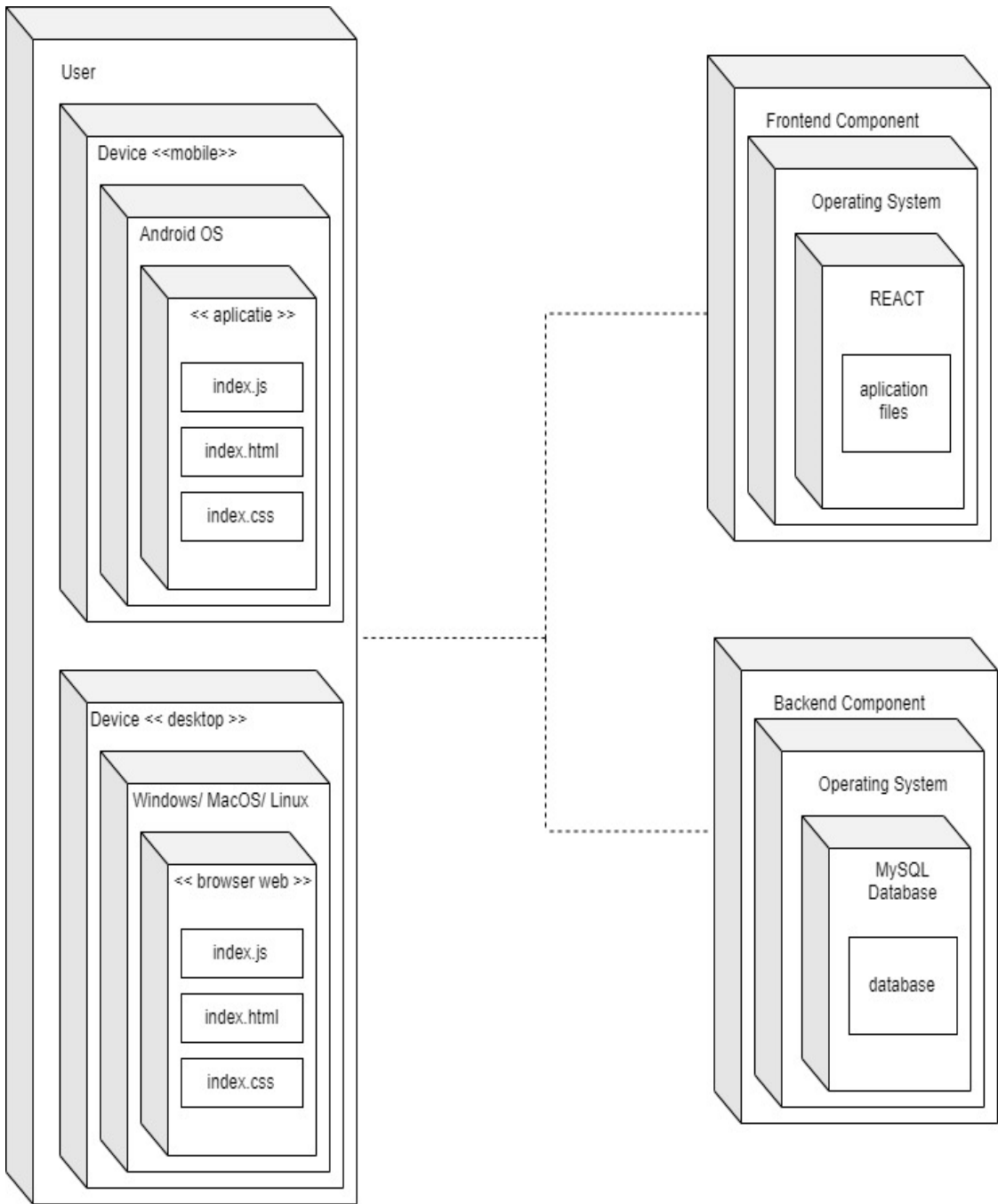
O arhitectură ierarhică închisă, în 3 niveluri este alcătuită din 3 tipuri de subsisteme:

1. Subsistemul Interfață utilizator:
 - a. ecran prin care utilizatorul interacționează cu sistemul
 - b. aduce datele ce vor fi afișate prin interacțiunea cu serverul web
 - c. trimite cereri de creare și modificare a datelor către server-ul web
2. Subsistemul Web server:
 - a. procesează evenimentele generate de interacțiunea cu utilizatorul
 - b. transmite mesaje la baza de date pentru accesarea sau modificarea datelor în conformitate cu interacțiunea utilizatorului
 - c. transmite mesaje către interfața utilizator pentru a actualiza design-ul prezentat pe ecran
3. Subsistemul Bază de date:
 - a. reprezentare a datelor specifice aplicației
 - b. operațiile de acces la date
 - c. operațiile de actualizare a datelor, conform interacțiunii cu utilizatorul și logicii aplicației

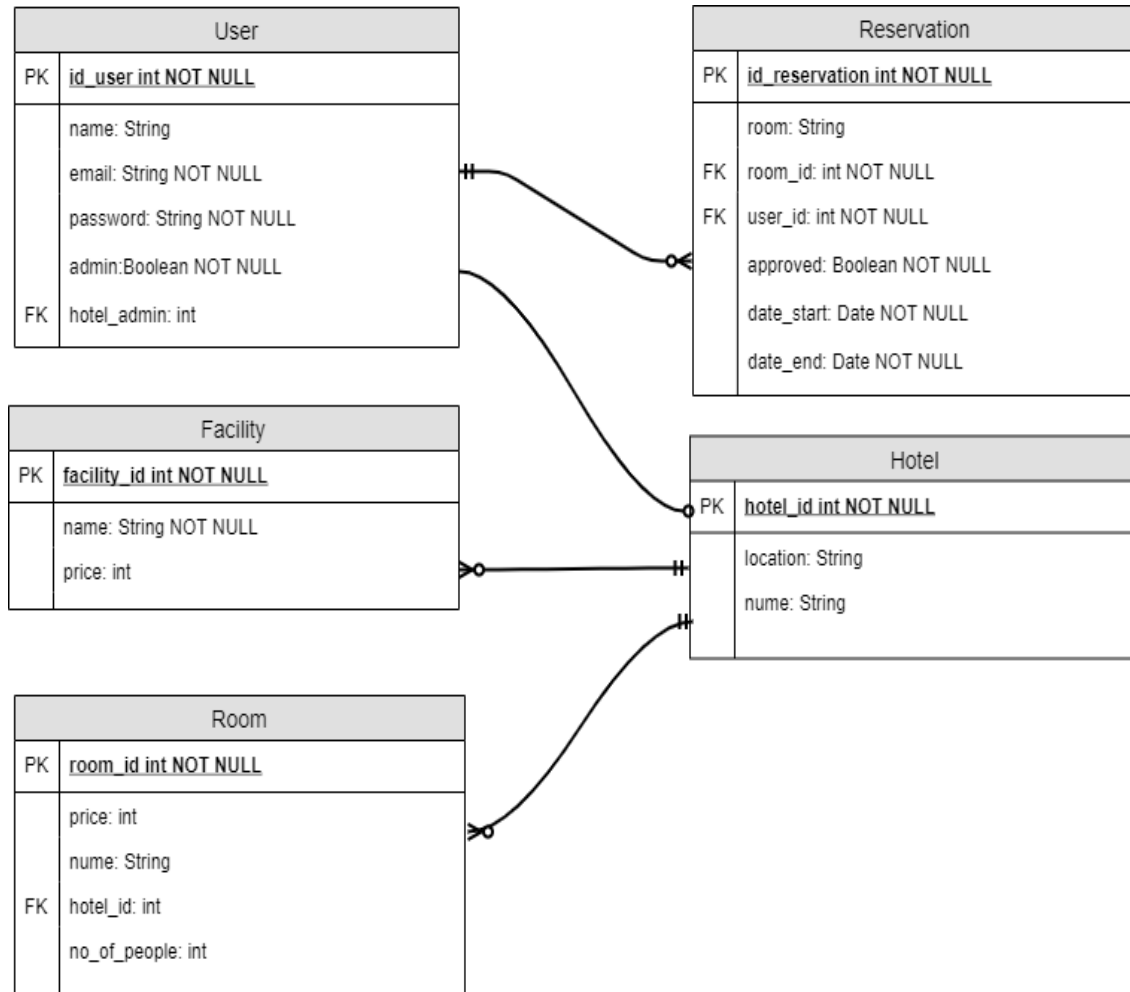
3.2 Descompunerea în subsisteme și responsabilitate fiecărui subsistem



3.3 Distribuția subsistemelor pe platforme hardware/software (diagrama de distribuție)



3.4 Managementul datelor persistente



Aplicația noastră va avea date care nu se distrug atunci când aplicația mobilă este închisă sau atunci când website-ul a fost închis.

Datele fiecărui utilizator pot fi regăsite și actualizate în cursul mai multor executări.

3.5 Controlul accesului utilizatorilor la sistem

Accesul utilizatorilor pe platforma se face prin înregistrare pentru clienți, iar după înregistrare își pot accesa contul cu credențialele folosite. Administratorii sunt introduși direct în baza de date, rezolvând astfel o problema de securitate reprezentată de un client ce ar avea privilegii de administrator. Autentificarea se face pe baza numelui de utilizator și a parolei.

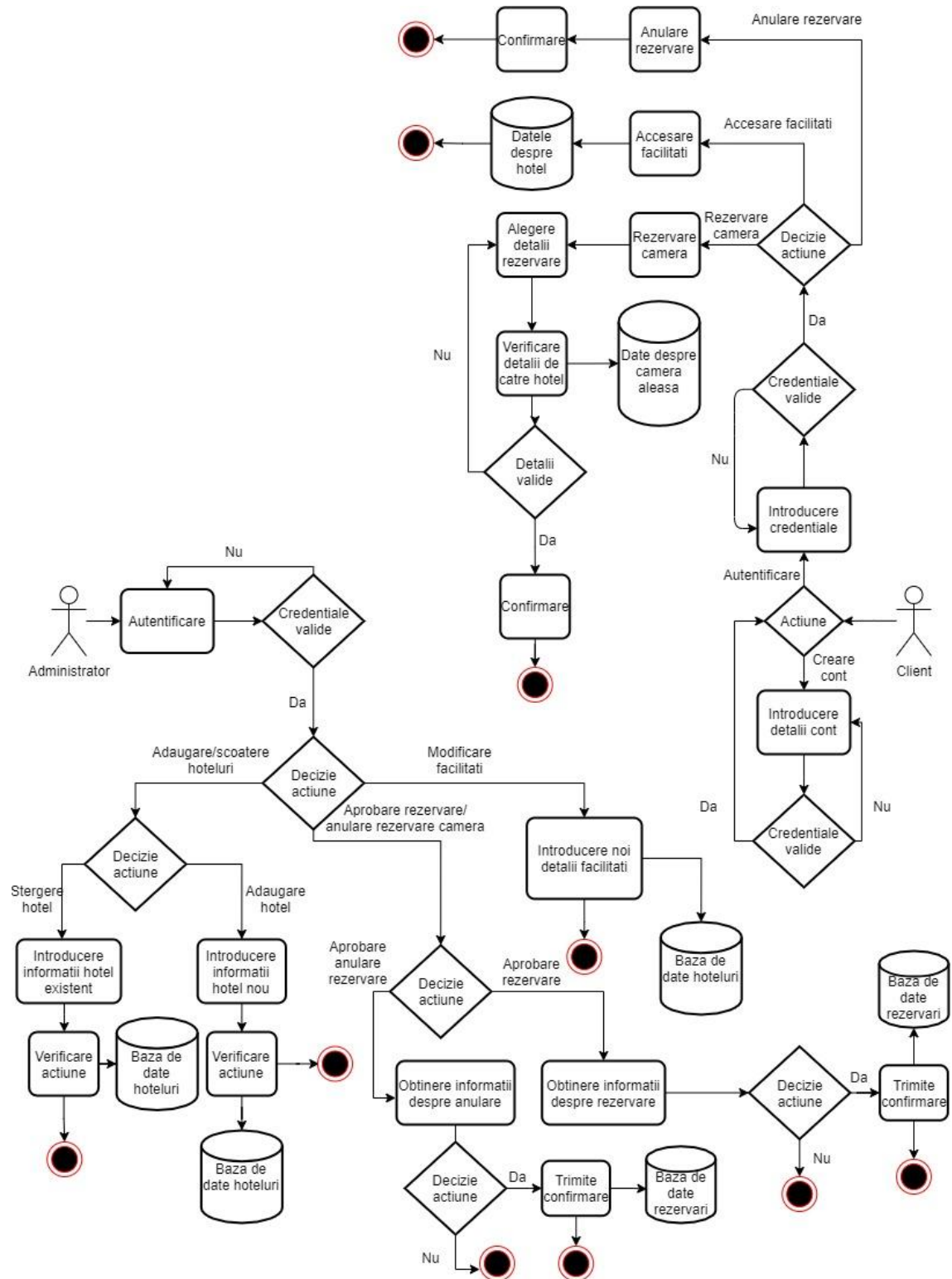
Utilizatorul client:

- Poate face rezervări în cadrul oricărui hotel disponibil din lanțul hotelier
- Poate vedea detalii despre diferitele tipuri de camere disponibile
- Poate vedea alte facilități oferite de hotelul ales

Utilizatorul administrator:

- Poate aproba/refuza o cerere de rezervare a unei camere
- Poate schimba perioada de disponibilitate a unei camere
- Poate adăuga/șterge facilități din cadrul unui hotel

3.6 Fluxul global al controlului



3.7 Condițiile limită (cazurile de utilizare limită)

Condițiile limita la care poate fi supus sistemul:

- Foarte multe query-uri către baza de date
- Întârzieri în comunicarea cu utilizatorul datorită supraîncărcării sistemului
- Gestionarea rezervărilor către aceleași camere și / sau aceleași facilități în același timp