

SISTEM DE PRELUARE COMENZI SI ORGANIZARE INTR-UN FAST FOOD

1. Prezentarea cerintelor functionale si non-functionale si identificarea acelor cerinte care influenteaza arhitectura

Cerinte functionale:

Sistemul trebuie sa permita clientilor accesul la informatii despre procesul de plasare a unei comenzi.

Sistemul trebuie sa permita clientilor sa initieze o comanda.

Sistemul trebuie sa permita clientilor sa plaseze o comanda.

Sistemul trebuie sa permita administratorului/barmanului/chelnarului sa preia comanda.

Sistemul trebuie sa permita comunicare administratorului/ barmanului/ chelnarului cu clientul oferind date despre faptul ca cererea a fost inregistrata si ora livrării comenzii .

Cerinte non-functionale:

Sistemul va fi implementat in Visual Studio 2017, cu ajutorul tehnologiilor NodeJs/Express + AngularJs 4

Baza de date implementata va fi de tip nerelationala (MongoDB)

Cerinte care influenteaza arhitectura:

Sistemul va fi implementat pe baza unei arhitecturi de tip client-server

2. Descompunerea in componente, definirea responsabilitatilor componentelor si a relatiilor dintre ele; argumentare

Sistemul va fi compus din 4 componente interconectate intre ele:

1. Aplicatia web:

Administratorul va avea drepturi totale

1. Adaugare oferte
2. Actualizare preturi

3. Preluare comenzi
4. Tranzactionare
5. Acceptare/respingere comanda
6. Adaugare noi meniuri

Barmanul/chelnerul/ospatarul

1. Preluare comenzi
2. Tranzactionare
3. Acceptare/respingere comanda

Client

1. Accesare informatii oferite despre meniuri si preturi
2. Plasare comanda
3. Vizualizare status comanda

2. Aplicatia pe telefon

Cienti

1. Accesare informatii oferite despre meniuri si preturi
2. Plasare comanda
3. Vizualizare status comanda

3. Baza de date nerelationala

4. Sistemul de autentificare si autorizare necesar atat clientilor cat si barmanilor, ospatarilor, chelnerilor, administratorului

3. Prezentare sistemului software din doua perspective (o diagrama pentru fiecare perspectiva + explicatii)

Prima perspectiva:

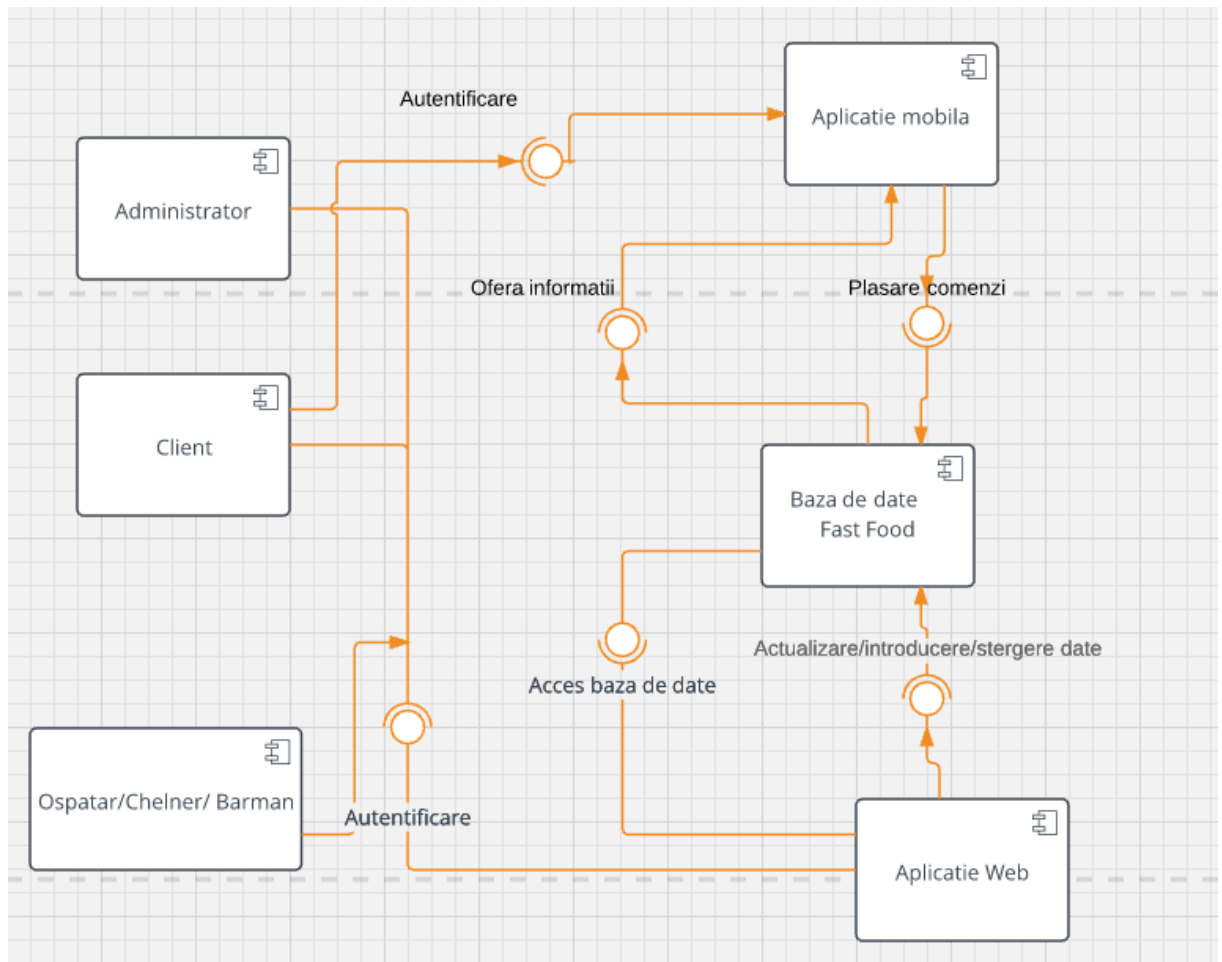


Figura 1 Diagrama de componente

Fiecare dintre Administrator, Client sau Ospatar/Chelner/Barman se poate autentifica in sistem prin intermediul interfetei de autentificare. Dupa ce au fost autentificati si autorizati fiecare va primi drepturi de acces in functie de rolul lui.

Baza de date comunica cu cele 2 aplicatii(Mobile si Web)

A doua perspectiva:

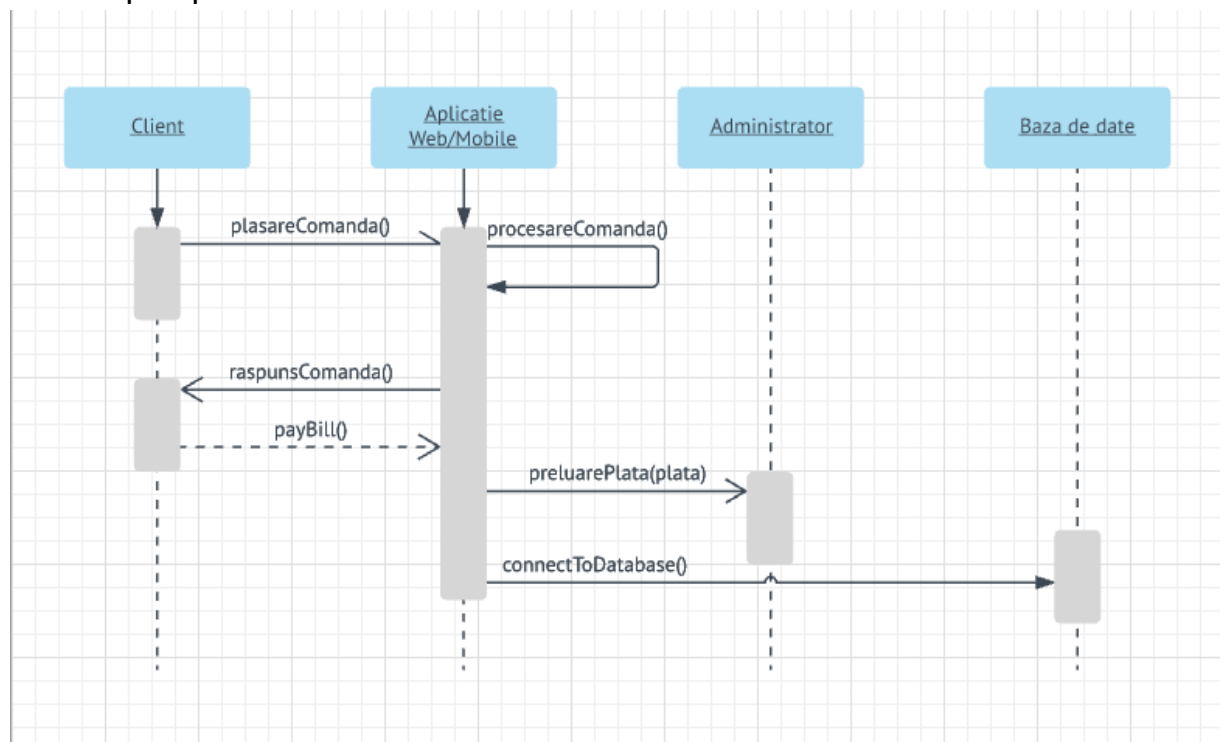


Figura 2 Diagrama de secventa

In prima faza clientul plaseaza o comanda, introducand datele necesare cerute de aplicatia web sau de aplicatia mobila. Comanda e procesata si apoi se va trimite raspunsul clientului.

Dupa ce masa a fost servita urmeaza efectuarea platii care e trimisa catre administrator si urcata in baza de date

Clientul poate opta pentru mai multe tipuri de plata(card, cash, bonuri)

4. Identificarea celor mai importanti 3 indicatori de calitate, specificarea masurii alese pentru fiecare indicator de calitate si argumentarea alegerii

1. Timpul de raspuns

Pentru un client este foarte important timpul in care primeste raspunsul la comanda. Timpul de raspuns va fi o problema atunci cand vor fi mai multe comenzi simultane, iar pentru aceasta va trebui sa suplimentam puterea de calcul prin adaugarea memoriei necesare cat si a procesorului corespunzator.

2. Securitatea informatiei transferate

Deoarece platile se pot face si direct prin aplicatia mobila sau aplicatia web va trebui sa asiguram un nivel de securitate ridicat pentru a nu fi furate date personale. Pentru aceasta este necesar ca schimbul de mesaje sa fie criptat si autorizat. Se va folosi tehnologia SSL/TLS

3. Portabilitatea

Fiecare client va dori sa acceseze aplicatia de pe dispozitivul care il are la indemana, iar de aceea va trebui ca tot sistemul sa fie compatibil cu diferite sisteme hardware, diferite dimensiuni de ecran si diferite versiuni de browsere sau sisteme de operare.

5. Identificarea tehnologiilor middleware folosite pentru a comunica intre componente, argumentarea alegerilor

Se va folosi modelul bazat pe livrarea mesajelor deoarece expeditorul trebuie să fie sigur că mesajul va ajunge la destinatar. Atunci când posesorul unei cărți de credit face o plată se poate ca acea plată să fie pusă într-o coadă urmând să fie procesată mai târziu. Dacă echipamentul hardware pe care este găzduită coada se defectează atunci dacă informația despre tranzacție nu este recuperată nu se va mai realiza plata către restaurant.

6. Identificarea principalelor modele si stiluri arhitecturale folosite, argumentarea alegerilor

Se va utiliza modelul client-server. Aplicatiile(web+mobile) vor fi clientii care vor avea acces, prin intermediul serverului, la baza de date a sistemului.

Clientii vor comunica cu serverul prin intermediul internetului, mai exact vor trimite si primi request-uri de tip HTTP.

Se va folosi pachetul MEAN care va face diferenta exacta intre aplicatii si care va putea fi testat deoarece modulele sunt bine definite.

7. Prezentarea scenariilor de validare a arhitecturii

Scenarii posibile:

1. Clientul plaseaza comanda. Validarea se va face prin faptul ca clientul va primi un raspuns de confirmare/infirmare la faptul ca comanda a fost preluata.
2. Clientul primeste comanda. Validarea se va face prin confirmare/infirmare din partea clientului
3. Clientul face plata. Validarea se va face prin primirea confirmarii/infirmarii a faptului ca plata a ajuns in atentie administratorului
4. Mai multi clienti comanda in acelasi timp. Validarea se va face prin gestionarea corecta a cererilor astfel incat sa nu se piarda nici o comanda.

Emanuel MOLDOVAN

An IV IS IP Gr.2.1