Sistem de control al unui restaurant

Anton Alexandru

an 4 IS

* prezentarea cerintelor functionale si non-functionale si identificarea acelor cerinte care influenteaza arhitectura

Cerinte functionale

-ofera clientilor accesul la ofertele din meniu si pretul acestora

-permite clientilor sa efectueze o comanda

- clientii au posibilitatea sa afle in ce stadiu se afla comanda, timpul ramas pana la servire

-permite bucatarilor si ospatarilor vizualizarea comenzilor

-posibilitatea modificarii si actualizarii automate a bazei de date a restaurantului

Cerinte non-functionale

-va fi implementat in xCode . cu ajutorul limbajului de programare Swift

-baza de data va fi implementata in Firebase

Cerinte care influenteaza arhitectura:

-sistemul va fi de tip client-server.

* Descompunere in component, definirea responasbilitatilor componentelor si a relatiilor dintre ele; argumentare

Aplicatie desktop 1:

Utilizator: bucatarul

* + - * Va putea vedea comenzile si prioritatea lor
      * Va putea modifca meniul in cazul in care nu se mai poate servi un anumit fel de mancare
      * Sunt obligati sa updateze starea preparatului

desktop 2:

Utilizator: managerul

* + - * Are o autentificare id si parola
      * Va putea introduce si sterge preparete din meniu

Aplicatie iphone/ipad:

Utilizator: clientul

* + - * Va putea vizualiza continutul meniului
      * Va selecta optiunile
      * Va trimite comanda

Ospatarul

* + - * Va vedea comenzile carei mese ii apartine
      * Va fi notificat cand este solicitat de client

O baza de date care contine denumirea produselor, ingredient si pret, iar la pornirea aplicatiei pe telefon se preiau datele din baza de date si se populeaza obiectele pentru a putea sa le afize in lista.

O alta baza de date care e pentru comenzi, la finalizarea comenzii se iau datele din comanda si se introduce in baza de date pentru bucatari, se adauga ca intr.o stiva si se sterge din vaza de date automat cand bucatarii termina comanda.

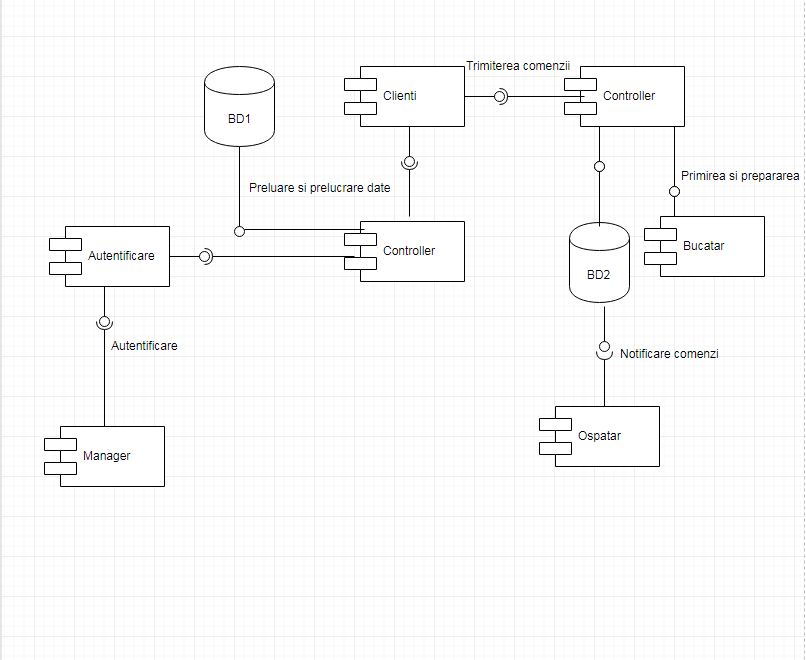
Schimbul de date este criptat.

O clasa care se ocupa cu gestionarea comenzilor.

Fiecare dispozitiv este configurat la instalarea aplicatiei nefiind necesara logare/delogare ulterioara.

Se folosesc socket-uri pentru trimiterea de date.

Prezentarea sistemului software din doua perspective



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

1. Timpul de raspuns

Avand un numar mare de clienti si un meniu destul de larg, timpul de raspuns al aplicatiei la o comanda noua trebuie sa fie cat mai mic, aplicatia sa fie mai eficienta.

1. Scalabilitatea

Sistemul trebuie sa raspunda la fel la numarul mare de comenzi si la adaugarea altor produse in lista de meniu.

1. Disponibilitatea

In cazul in care el nu functionaza sa fie un chelner la dispozitie pentru a prelua el comanda.

Identificarea tehnologiilor middleware folosite pentru a comunica intre component, argumentarea alegerilor:

- se folosesc cozi cu mesaje in cazul in care comenzile sunt plasate la cateva secunde dintanta una de cealalta, iar controlerul la momentul primirii comenzii o introduce in baza de date fiind doar o instructie simpla de tipul insert iar la finalizarea comenzii o sterge imediat din baza de date fiind doar instructiune simpla de delete

Identificarea principalelor modele si stilurilor arhitecturale folosite, argumentarea alegerilor

- este folosit modelul client-server si modelul arhitectural Passive View , deoarece toate cererile utilizatorului sunt preluate de un controller, in rest el este pasiv.

Prezentarea scenariilor de validare a ahritecturii:

-clientul plaseaza comanda-ajunge la controler si o insereaza in baza de date

-bucatarul primeste comanda-o sterge din baza de date

- mai multi clienti insereaza comenzi simultan- se introduc intr.o stiva comenzile iar controller-ul le insereaza pe rand

-managerul introduce un produs nou-controllerul se insereaza in baza de date si cand sterge un produs se sterge