

Documentul de proiectare arhitecturala Architectural Design Document (ADD)

1. Introducere	2
1.1 Scopul sistemului	2
1.2 Definiții, acronime	2
1.3 Documente referințe	2
2. Obiective de proiectare	2
3. Arhitectura propusă	3
3.1 Prezentare generala a arhitecturii sistemului	3
3.2 Decompozitia în subsisteme și responsabilitățile fiecărui subsistem	3
Diagrama de componente:	3
Descrierea interfețelor:	4
Clase:	5
3.3 Distribuția subsistemelor pe platforme hardware/software	6
3.4 Managementul datelor persistente	6
3.5 Controlul accesului utilizatorilor la sistem	8
3.6 Fluxul global al controlului	8
3.7 Condițiile limita	9
Planificarea implementării proiectului	10

1. Introducere

1.1 Scopul sistemului

Acest document are rolul de a oferi o descriere amplă asupra proiectului PoliFood, mentionand obiectivele de proiectare si arhitectura propusă care cuprinde împărțirea în subsisteme și distribuția acestora pe platforme hardware/software, gestiunea datelor persistente, controlul accesului utilizatorilor la sistem, fluxul global al controlului și condițiile limita.

1.2 Definiții, acronime

UPB = Universitatea Politehnica din București

Utilizatori simpli = Studenții și angajații UPB

Utilizator Special = Administrator Aplicație

MySql = sistem gratuit și open-source de baze de date relaționale

Spring = framework de creare de programe Java, bazat pe principiul inversării controlului

Hibernate = framework Java care definește interacțiuni cu sisteme de baze de date, în baza Java Persistence API

1.3 Documente referințe

Documentul de specificarea cerințelor:

https://docs.google.com/document/d/19tVRCYQdjoKXNeJ2Z1vX8i1rUMHEB_QZmXm6oMLYoMQ/edit#heading=h.8ih8ktp04wda

2. Obiective de proiectare

Un prim obiectiv de proiectare îl reprezintă accesibilitatea aplicației. Fiind o aplicație web, ea poate fi accesată dintr-un browser de pe orice dispozitiv cu acces la Internet. În acest scop, vom folosi Spring pentru componenta de backend. Pentru a asigura disponibilitatea aplicației, vom avea un server care va rula continuu pe o mașină virtuală de cloud și va hosta componentele frontend și backend.

Ne propunem ca aplicația să aibă și o interfață ușor de folosit și intuitivă. Pentru aceasta vom folosi Vue.js și/sau Bootstrap. Interfața va consta dintr-un meniu minimalist aflat în partea stângă a ecranului. Fiecare element al meniului (setări, vizualizare rezervări, vizualizare cantine etc.) va

redirecționa utilizatorul către pagina corespunzătoare de unde va putea continua interacțiunea cu acesta, putând oricând schimba pagina din meniu.

Stocarea datelor utilizatorilor se va face printr-o baza de date care va fi implementată prin MySQL și vom interacționa cu aceasta prin intermediul Hibernate. Baza de date va conține atât datele de autentificare ale utilizatorilor, cât și meniurile rezervate de aceștia. Nu vom reține alte date personale ale utilizatorilor în afara de numele acestora folosit pentru identificarea lor.

Din punct de vedere al securității, aceasta va fi asigurată prin folosirea protocolului https. Pentru criptarea parolelor și a traficului https vom folosi Spring Security. Certificatul site-ului va fi generat prin serviciul Let's Encrypt.

Datele personale preluate de aplicație vor fi criptate, iar pentru a preveni abuzurile de serviciile oferite de aplicație, vom limita numărul de meniuri ce pot fi rezervate. În plus, plata meniurilor se va face fizic la cantina, astfel ea nu reprezintă o problemă de securitate.

3. Arhitectura propusă

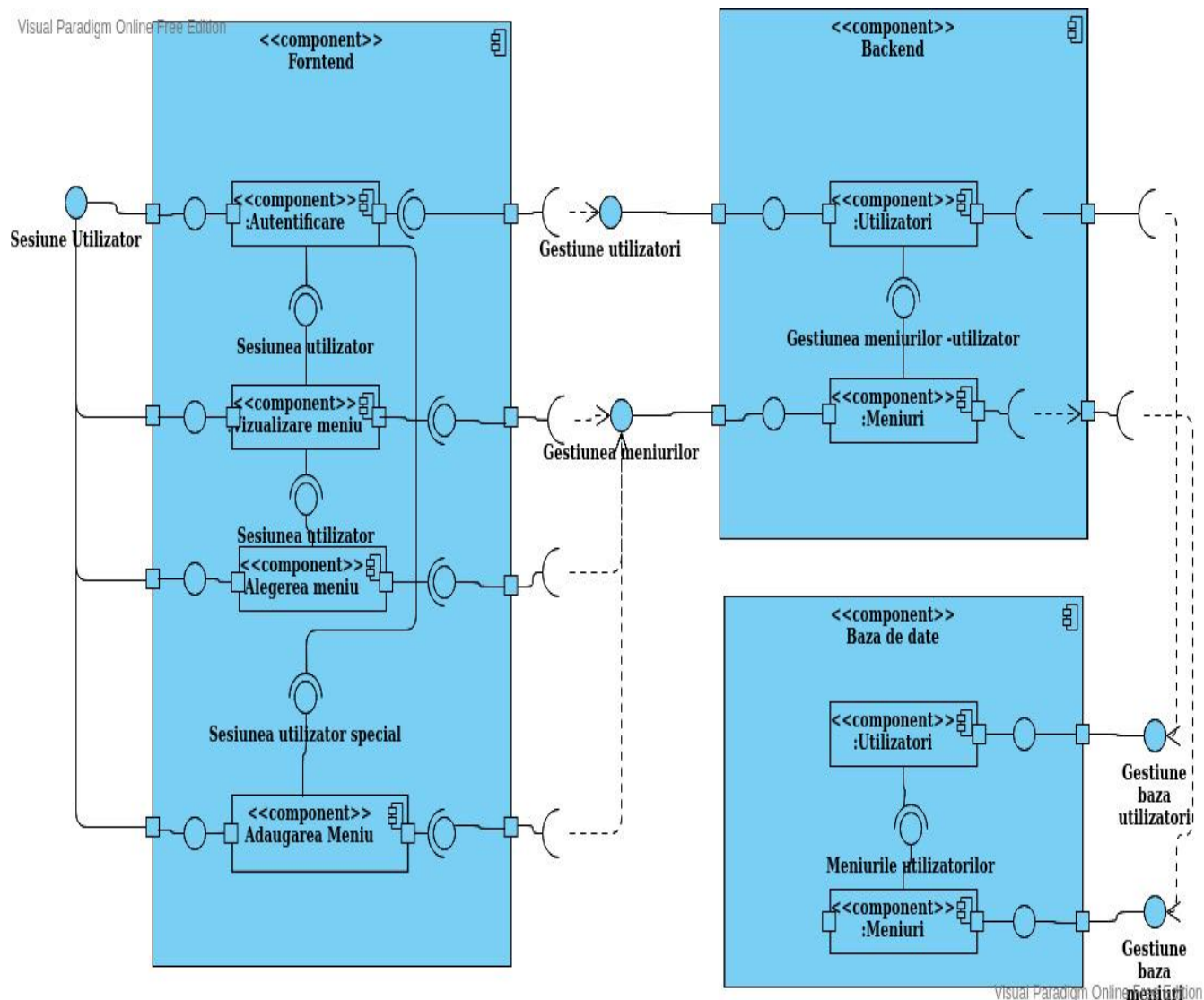
3.1 Prezentare generală a arhitecturii sistemului

Arhitectura sistemului constă în 3 subcomponente care comunică între ele: Frontend, Backend și Baza de date. Componenta Frontend are rolul de a pune la dispoziția utilizatorului o interfață accesibilă și intuitivă cu ajutorul căreia să poată rezerva mâncarea sau să modifice diferite aspecte ale contului. Baza de date este folosită pentru gestiunea utilizatorilor, reținând atât detalii de autentificare, cât și meniurile rezervate. Legătura dintre Frontend și Baza de date o face componenta Backend, care va interoga și modifica baza de date conform cererilor utilizatorilor.

3.2 Decompoziția în subsisteme și responsabilitățile fiecărui subsistem

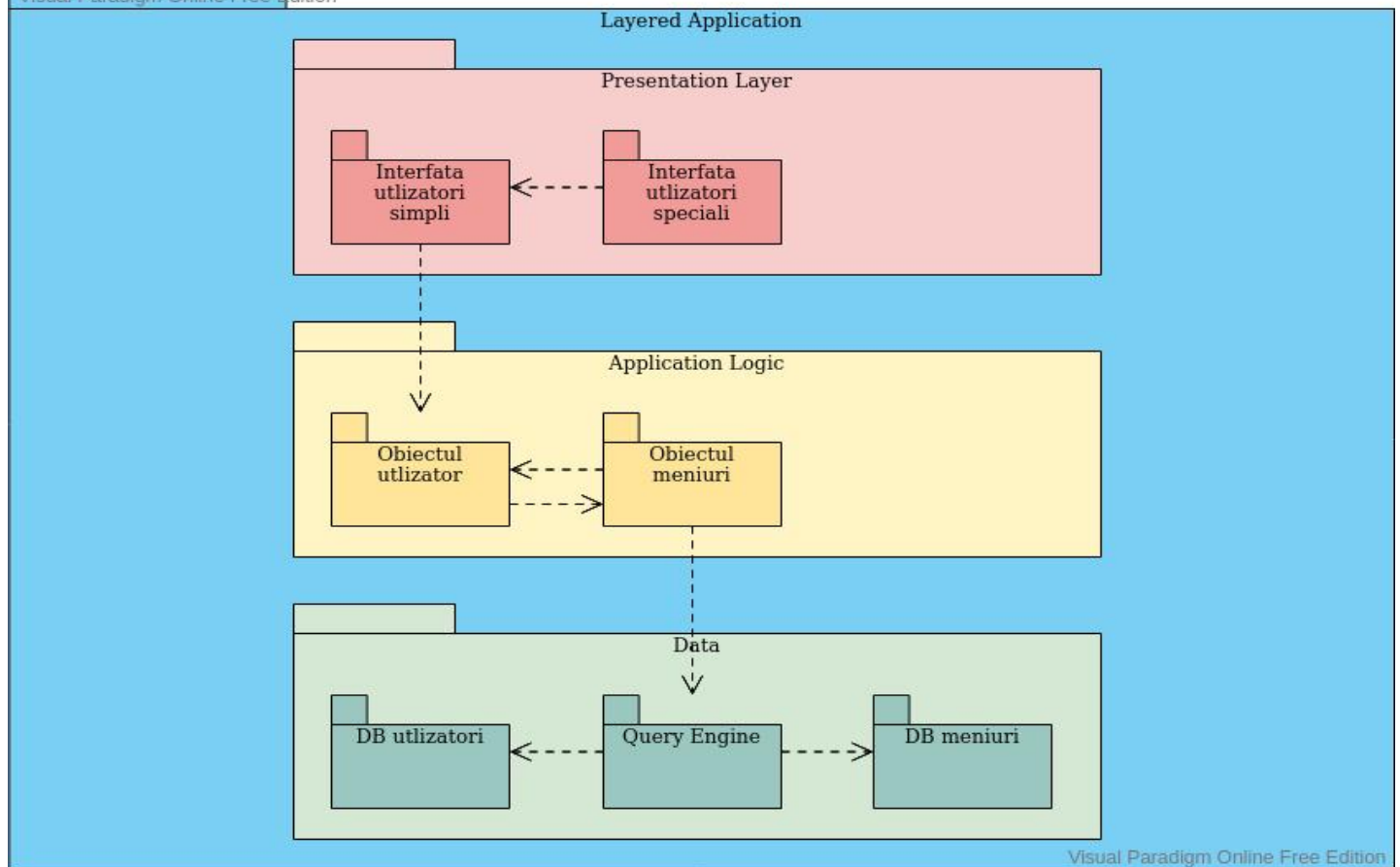
Diagrama de componente:

În această diagramă sunt 3 componente: partea de frontend care face legătura cu clienții printr-o pagină web. Toate apelurile făcute în partea de frontend sunt trimise către backend, care parsează cererea și dacă sunt necesare date din partea 3, bază de date, trimite o cerere către baza de date

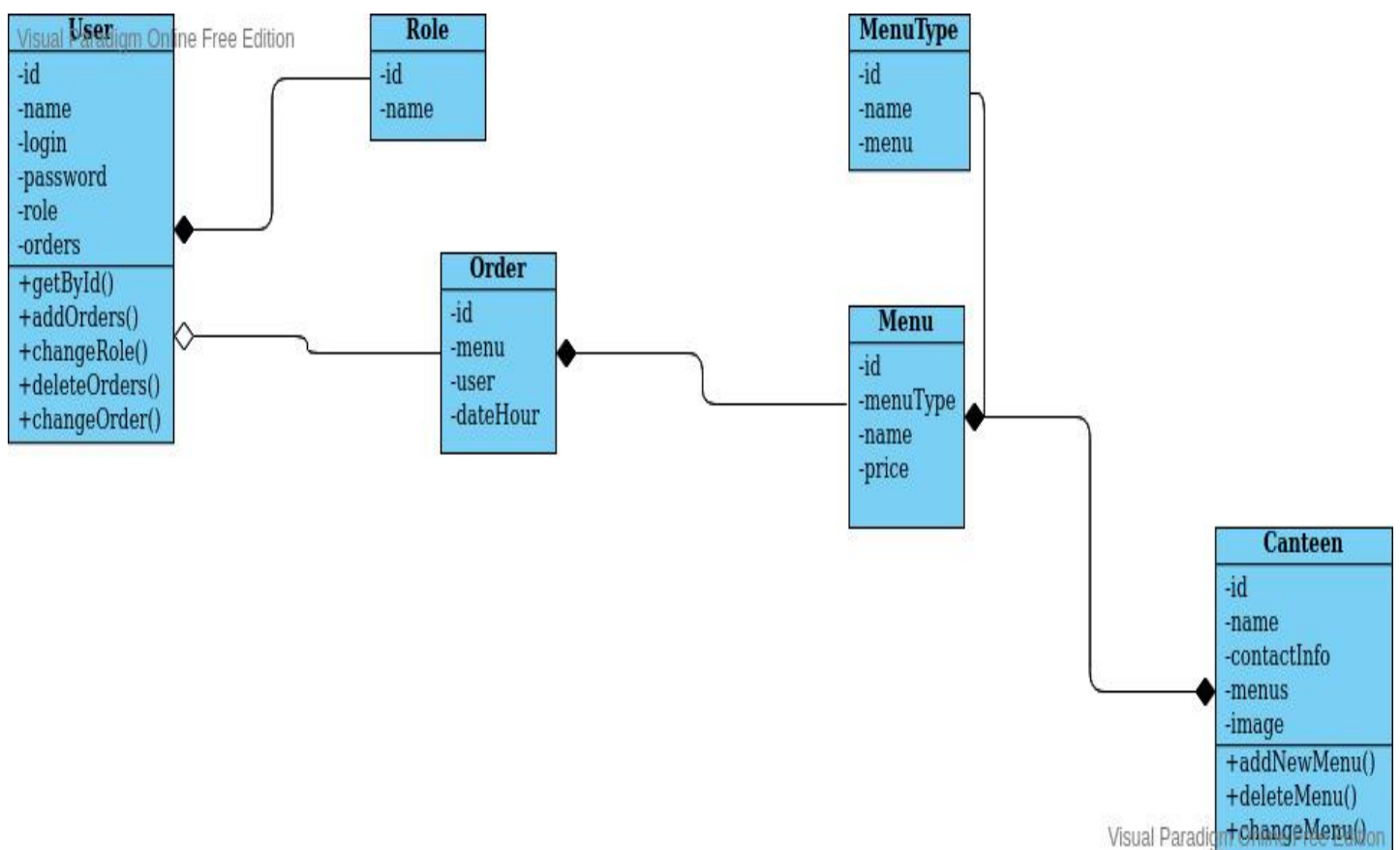


Descrierea interfețelor:

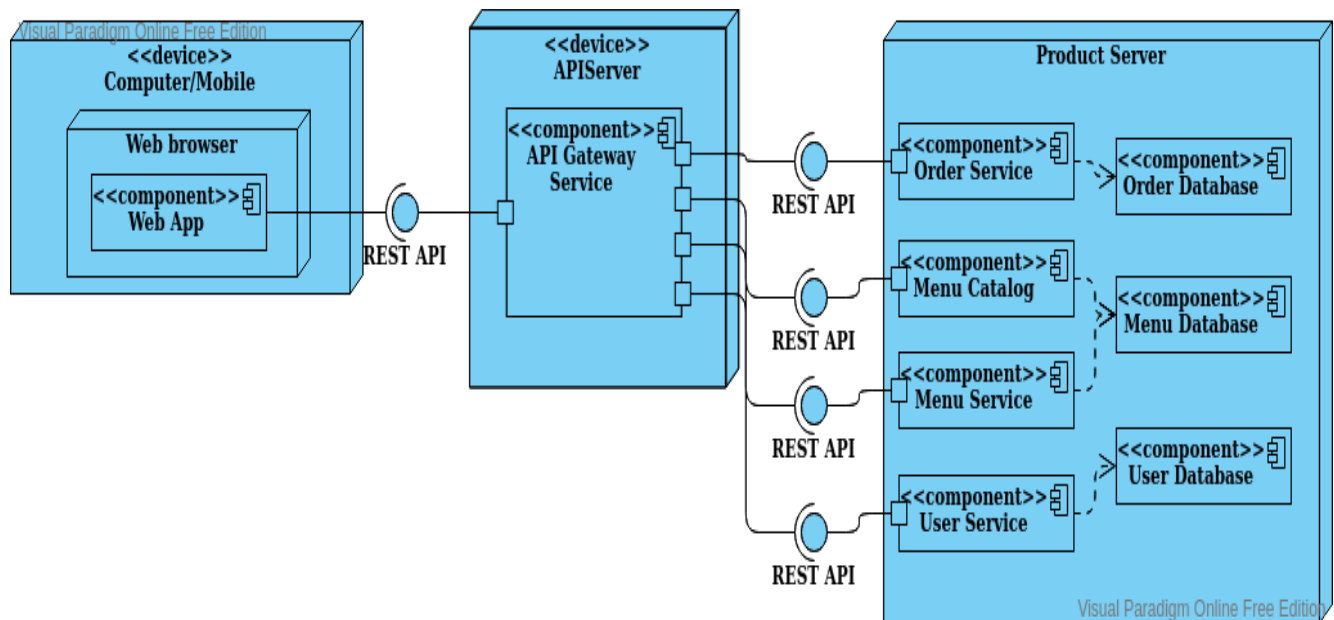
La fel ca mai sus, și aici sunt 3 componente, în partea de prezentare, fiecare utilizator va avea forma lui de pagina web, în dependență de rolul care îl are. Utilizatorii speciali pot fi și utilizatori simpli. Toată logica aplicației se face în java Spring, unde 2 componente mari și importante sunt utilizatorii și meniurile, a căror date sunt luate din baza de date, din a 3 componenta prin diferite interogari.



Clase:



3.3 Distribuția subsistemelor pe platforme hardware/software



3.4 Managementul datelor persistente

Managementul datelor persistente se va efectua printr-un server MySQL și pachetul de interfatare Hibernate MySQL din framework-ul Spring.

Utilizatorii sunt stocați în tabela User. Aceștia au un singur rol și mai multe comenzi. Cantinele sunt stocate în tabela Provider și au mai multe tipuri de mancare disponibile, stocate în tabela Menu. Legatura Many to Many dintre Order și Menu se face prin tabela de legatura OrderItem.

User

id	numeric	PK
email	text	
password	text	
firstName	text	
lastName	text	
accountCreationDate	date	
roleId	numeric	FK

Role

id	numeric	PK
name	text	

Order

id	numeric	PK
date	date	
ordered	tinyint(1)	
userId	numeric	FK

OrderItem

id	numeric	PK
price	float	
orderId	numeric	FK
menuId	numeric	FK

Menu

id	numeric	PK
providerId	numeric	FK
name	text	
dayOfWeek	text	
enabled	tinyint(1)	

MenuType

id	numeric	PK
name	text	
menuId	numeric	FK

Provider

id	numeric	PK
name	text	
description	text	
enabled	tinyint(1)	

3.5 Controlul accesului utilizatorilor la sistem

Sistemul are 3 tipuri de utilizatori: clienți, angajați ai cantinei și administratori ai platformei. Accesul se face prin aceeași pagină de login, iar în funcție de rolul lor ei primesc altă pagină principală.

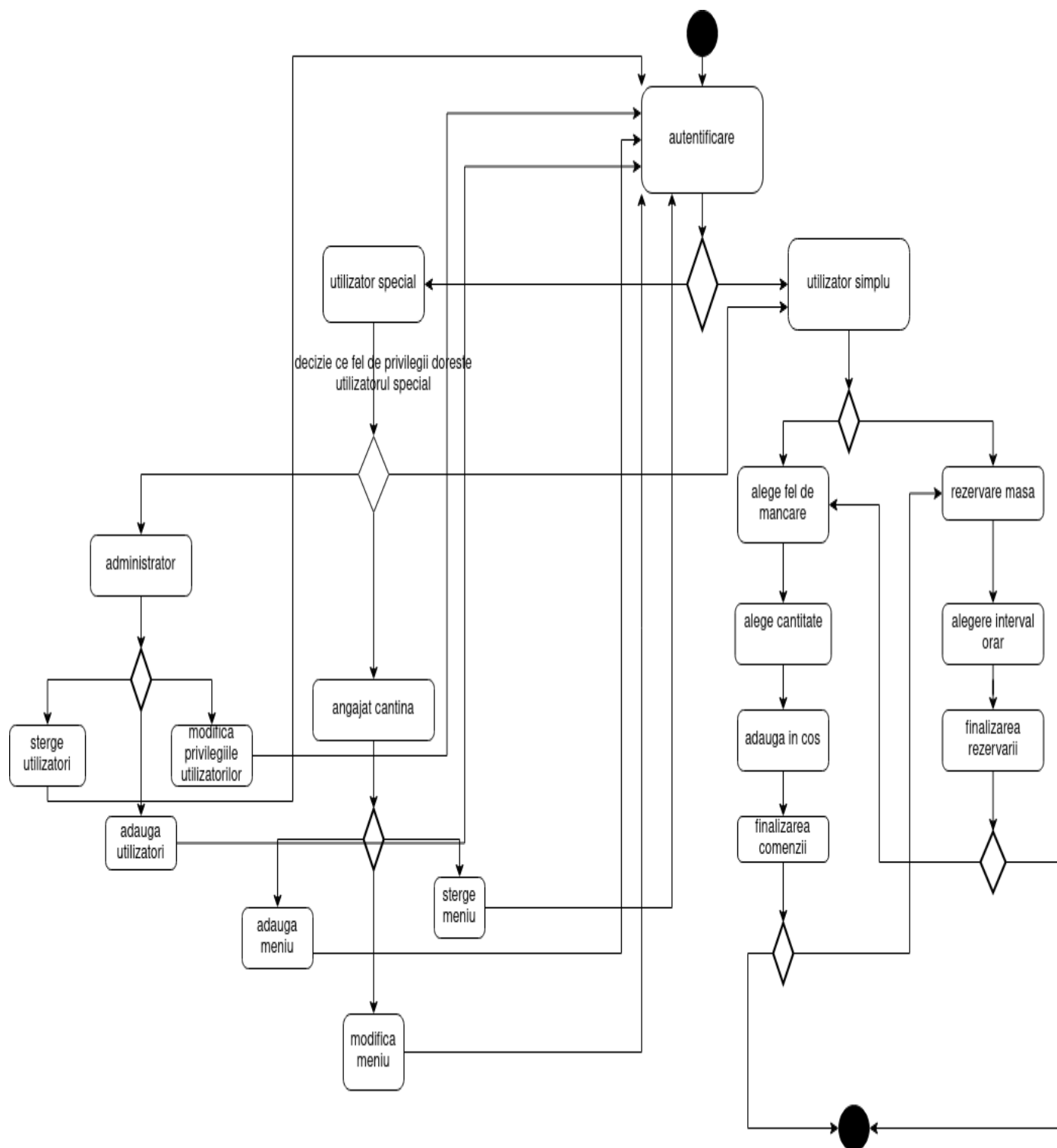
Pentru a se păstra separarea conturilor, se verifică faptul că utilizatorul are acces la o anumită pagină atunci când face request-ul către server-ul de Backend.

3.6 Fluxul global al controlului

Fluxul global al controlului se va face prin așteptarea unui input de la utilizator. Utilizatorul se va loga/autentifica, după care va fi redirectionat către pagina principală. Aici depinde de fiecare utilizator simplu ce își dorește să adauge în coș. De pe pagina principală se va oferi input prin apăsarea unor butoane care vor semnifica produsul dorit și cantitatea dorită. Acestea se vor adăuga în coșul de cumpărături, de unde utilizatorul va comanda produsul dorit.

Pe lângă produsele comandate, utilizatorul poate opta pentru alegerea de a rezerva o masă în cadrul cantinei într-un anumit interval orar.

Utilizatori speciali pot alege dacă să se comporte ca un utilizator simplu sau pot continua pe platformă ca un utilizator cu privilegii și pot modifica/șterge/adauga meniurile.



3.7 Condițiile limita

Condiția limită a platformei o reprezintă faptul că prea mulți utilizatori se pot conecta în același timp pe platformă și platforma poate ceda. În acest caz vom scoate o perioadă platforma din funcționare până rezolvăm problemele de mentenanță, după care va fi din nou disponibilă.

Planificarea implementării proiectului

ID	Nume	Asignat	Sprint	Estimare
S-1	Backend 1 și Baza de date	Toata echipa	1	30 h
S-2	Backend 2 și deployment initial	Toata echipa	2	1 saptamana
S-3	Frontend	Toata echipa	3	1 saptamana
S-4	Deployment final și Task-uri rămase	Toata echipa	4	4 zile

Sprint 1			
ID	Nume	Asignat	Estimare
A-1	Instalare mediu de dezvoltare	Toata echipa	1h
A-2	Creare repository Git și echipa pe GitHub	Damian Stefania-Dorina	1h
A-3	Scriere configurare Maven	Josan-Gulica Stela	2h
A-4	Instalare dependențelor necesare proiectului	Toata echipa	1h
BD-1	Conexiunea cu Backend	Melinte Paul-Eduard	1h
BD-2	Creare tabele	Ion Luminita-Andreea	2h
BD-3	Scriere query-uri	Ion Luminita-Andreea	2h
BD-4	Populare tabele pentru testare	Ion Luminita-Andreea	2h
BE-1	Creare obiectelor Entity și Repository	Damian Stefania-Dorina	4h
BE-2	Creare obiecte Controller	Josan-Gulica Stela	3h
BE-3	Scriere Endpoint-uri REST pentru utilizatori	Damian Stefania-Dorina	3h
BE-4	Scriere Endpoint-uri REST pentru cantine	Josan-Gulica Stela	3h
BE-5	Testare unitară	Melinte Paul-Eduard	3h
BE-6	Testare integrala	Melinte Paul-Eduard	2h