



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

DOCUMENTATIE

Tema 2: UTILIZAREA APLICATIEI DE
GESTIUNE A COZILOR, FIRE SI
MECANISME DE SINCRONIZARE

Nume: Grasu Elena-Lorena

Grupa:30222

An:2

Cuprins

1. Obiectivul temei
2. Analiza problemei, modelare, scenarii, cazuri de utilizare
3. Proiectare
4. Rezultate
5. Concluzii
6. Bibliografie

1. Obiectivul temei:

Tema abordata isi propune sa implementeze o aplicatie menita sa controleze fluxul cozilor in care sunt plasati o serie de clienti, in cadrul unui sistem informatic pentru a minimiza timpul de asteptare. Cu ajutorul limbajului de programare: Java , scopul sistemului este de a imbunatati timpii de raspuns si de a evita posibilele conflicte.

2. Analiza problemei , modelare , scenarii , cazuri de utilizare

Problema de fata doreste gestionarea eficienta a cozii de clienți, astfel încât să se diminueze timpul de așteptare. In cazul modelarii , în aplicație, va exista o coadă de clienți în așteptare în care fiecare cerere de serviciu va fi afișată în ordinea sosirii (ordonarea crescatoare a vectorului de task-uri). Se vor trata cererile în coadă în paralel cu threaduri, ceea ce permite procesarea mai multor cereri simultan.

La nivelul scenariilor si a cazurilor de utilizare pentru coada de clienți, putem avea:

Clienții pot solicita servicii prin telefon sau accesând sistemul, sau fizic completand pe loc informatiile necesare eliberarii unui bilet cu numar(ex: cozi de asteptare ANAF). În ordinea sosirii, cererile clienților sunt aranjate în coadă. Fiecare cerere va fi legată de un identificator unic, precum și informații despre serviciul pe care îl solicită solicitantul; In cadrul procesarea cererilor simultan, se vor crea fire de execuție pentru a gestiona cererile în așteptare. Fiecare fir de execuție va prelua o cerere din coadă și va oferi serviciul dorit. Procesarea mai multor cereri simultan va fi posibilă prin utilizarea firelor de execuție, ceea ce va reduce timpul total de așteptare.

La nivelul prevenirii conflictelor de concurenta , vor fi utilizate mecanisme de sincronizare. Aceste mecanisme vor preveni blocarea sau coruperea

datelor prin asigurarea că doar un fir de execuție poate accesa o cerere din coadă la un moment dat.

3. Proiectare

Pe baza principiilor de programare orientată pe obiecte (OOP) , Java este limbajul de programare folosit pentru a rula aplicația. Conceperea acestei aplicații a fost posibilă prin intermediul celor trei pachete: Model , bussinesLogic și gui.

În cazul primului pachet regăsim clasele atribuite obiectivelor principale ale acestei aplicații:

- a. Clasa “Server” reprezintă entitatea care gestionează cozile de cereri de la clienți și folosește fire de execuție pentru a le procesa.
- b. Clasa “Task” este utilizată pentru a încapsula informațiile necesare pentru procesarea cererilor din coadă, astfel încât firele de execuție să le poată procesa.

În cazul celui de-al doilea pachet sunt ilustrate clasele menite să mențină logica proiectului:

- a. Clasa “ConcreteStrategyTime” va dezvolta un plan specific pentru a calcula timpul de așteptare al clienților în cozi pe baza unor criterii specifice și de a realiza o metodă de alegere a cozii după timpul minim de așteptare
- b. Clasa “Scheduler” asigură o gestionare optimă a resurselor disponibile și de a oferi oportunitatea de a trata cererile în ordine optimă, astfel încât să se reducă timpul de așteptare și să se maximizeze eficiența.
- c. Clasa “SelectionPolicy” definește strategiile de selecție a următoarei sarcini (cereri) din coadă pentru a fi procesată. Aceasta va oferi un mod modular și flexibil de a implementa diverse politici de selecție, astfel încât să se poată schimba comportamentul de selecție fără a afecta restul aplicației.

- d. Clasa “ShortestQueueStrategy” cu cel mai mic număr de sarcini pentru a minimiza timpul de așteptare și a echilibra sarcinile în cadrul sistemului.
- e. Clasa “SimulationManager” este responsabilă de gestionarea întregului proces de simulare a sistemului de cozi și de coordonarea diferitelor componente ale aplicației.
- f. Clasa “Strategy” definește strategiile generice utilizate în gestionarea coziilor sau a altor procese din aplicație. Aceasta permite definirea diferitelor comportamente și algoritmi pentru acțiunile specifice, precum planificarea sarcinilor, selecția coziilor sau alte operații relevante.

Cel de-al treilea pachet contine clasele care fac posibila interfata utilizatorului:

- a. Clasa “ SimulationFrame” permite utilizatorilor să vizualizeze și să interacționeze cu simularea sistemului de cozi și să observe evoluția proceselor în timp real.

4.Rezultate

La nivelul rezultatelor , s-a obtinut o aplicatie care a ajutat la imbunatatirea aspectelor precum:

- a. Reducerea timpului de așteptare: Prin utilizarea unei strategii eficiente de selecție și planificare a sarcinilor în cozi, timpul de așteptare pentru clienți poate fi diminuat semnificativ. Acest lucru duce la o îmbunătățire a experienței clienților și la o creștere a satisfacției lor.
- b. Echilibrarea sarcinilor: Dacă aplicația este configurată pentru a folosi strategii de distribuție echilibrată a sarcinilor către cozi, se poate obține o distribuție uniformă a muncii în cadrul serverului și evitarea supraîncărcării unei singure cozi. Aceasta contribuie la evitarea situațiilor în care unele cozi sunt aglomerate în timp ce altele rămân subutilizate.

- c. Eficiența proceselor: O aplicație bine implementată poate duce la o gestionare eficientă a coziilor și a sarcinilor, ceea ce se traduce într-o eficiență sporită a întregului sistem. Procesarea sarcinilor poate fi realizată mai rapid și cu mai puține resurse.
- d. Monitorizarea și analiza: O aplicație de acest fel poate oferi date detaliate despre performanța sistemului, timpul de așteptare în cozi, numărul de clienți servit, etc. Aceste informații pot fi folosite pentru analize ulterioare și pentru optimizarea sistemului în funcție de nevoile reale.
- e. Flexibilitate: Prin utilizarea unor strategii și politici configurabile, aplicația poate fi adaptată pentru a se potrivi diferitelor scenarii și cerințe ale utilizatorilor.

5. Concluzii

În cele din urmă, aceasta aplicație este menită să simuleze cozi care preiau un anumit număr de sarcini pe care trebuie să le prelucreze folosind o strategie selectată.