CERINTA TEMA 1 curs - Analiza unui sistem software la alegere

---

- prezentarea cerintelor funcționale si non-functionale si identificarea acelor cerinte care influenteaza arhitectura
- descompunere in componente, definirea responsabilitatilor componentelor si a relatiilor dintre ele; argumentare
- prezentarea sistemului software din doua perspective (o diagram pentru fiecare perspectiva + explicatii)
- identitifcarea celor mai importanți 3 indicatori de calitate, specificarea masurii alese pentru fiecare indicator de calitate si argumetarea alegerii
- identificarea tehnologiilor middleware folosite pentru a comunica intre componente, argumentarea alegerilor
- identificarea pincipalelor modele și stiluri arhitecturale folosite, argumentarea alegerilor
- prezentarea scenariilor de validare a arhitecturii

SOLUŢIE PROPUSĂ PENTRU TEMA 1 curs - Analiză unui sistem software la alegere

SISTEM DE CONTROL AL CAZĂRII STUDENŢILOR ÎN CĂMINELE STUDENŢEŞTI

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

Cerinţe funcţionale:

Sistemul trebuie să permită studenţilor accesul la informaţii referitoare la procesul de cazare/lichidare, la documentele necesare(template contracte, cereri de grupare, cereri de lichidare, acte necesare), informaţii actualizate în timp.

Sistemul trebuie să permită studenţilor să iniţieze o cerere pentru cazare/lichidare.

Sistemul trebuie să permită studenţilor să completeze online documentele necesare pentru cazarea în cămin, să existe posiblitatea că actele cerute să fie apoi încărcate online.

Sistemul trebuie să permită administratorilor de cămin să actualizeze baza de date pentru propriul cămin(să adauge informaţii despre cămin, să încheie contracte cu studenţii, să şteargă studenții din baza de date în cazul lichidării).

Sistemul trebuie să permită administratorilor cămin să genereze la cerere liste, rapoarte cu situația pentru căminul propriu.

Sistemul trebuie să acorde drepturi totale administratorului de sistem(poate adaugă cămin nou în sistem, poate vizualiza situaţia căminelor, trebuie să verifice identitatea studenţilor care îşi creează un cont, preia informaţii legate de mediile studenţilor din baza de date a facultăţii<doar drept de read>, poate genera automat rapoarte).

Cerinţe non-funcţionale:

Sistemul va fi implementat în Visual Studio 2017, cu ajutorul tehnologiei ASP.NET MVC.

Sistemul va fi unul transparent pentru toţi utilizatorii. Fiecare utilizator poate vizualiza situaţia finală a aşezării în cămin(locurile din căminul propriu pentru studenţii şi administratorii de cămin).

Baza de date implementată va fi de tip Entity Framework.

Cerinţe care influenţează arhitectura:

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

- **descompunere in componente, definirea responsabilitatilor componentelor si a relatiilor dintre ele; argumentare**

Sistemul va fi compus din 7 componente interconectate între ele:

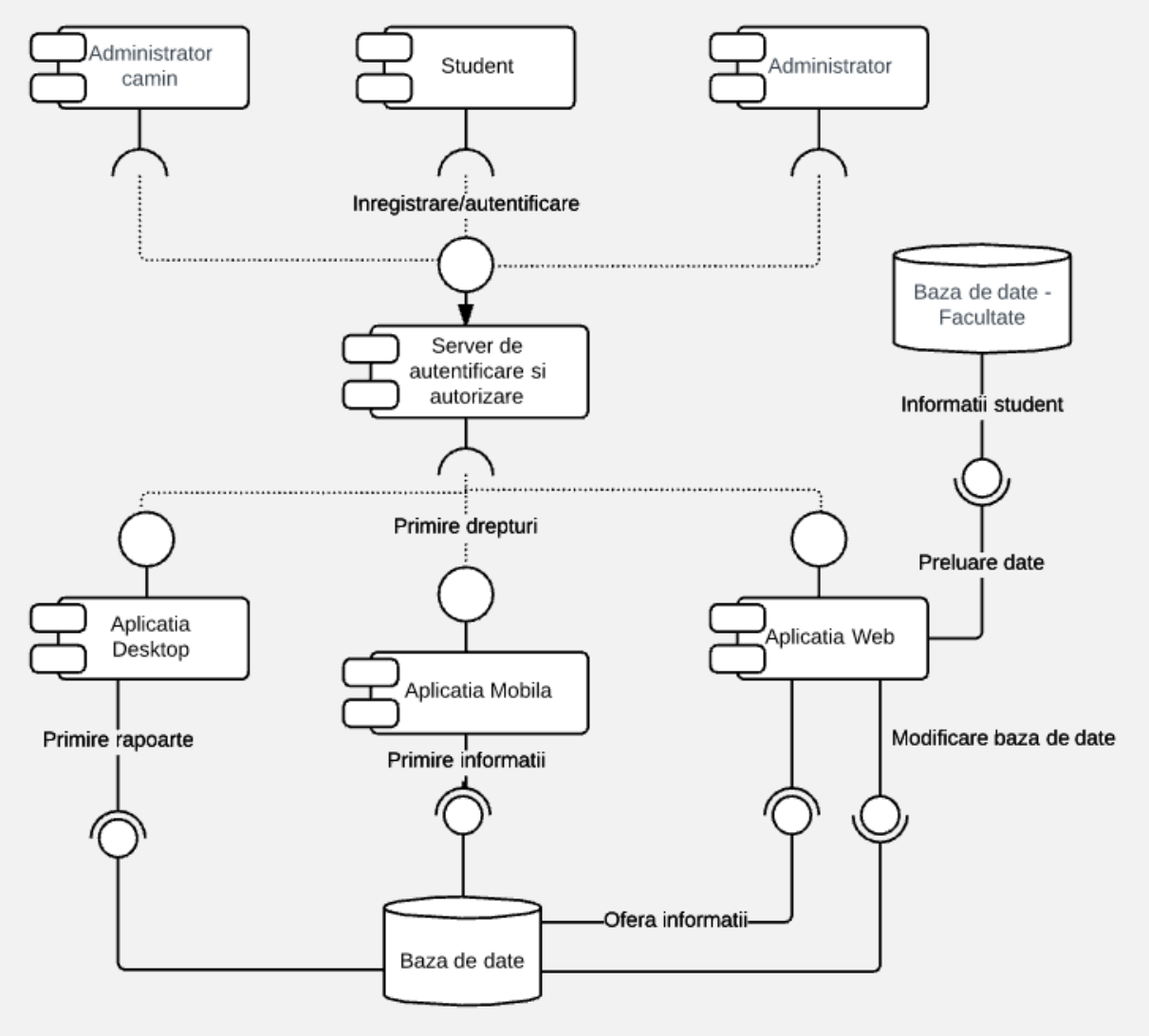
1) Aplicatia web
    a. Utilizatori – drepturi – cont pentru fiecare utilizator
        i. Admin – all rights
            1. Adăugare informații camin nou
            2. Generare automată de rapoarte, liste de interes
            3. Vizualizare situație cămine
            4. Preluare informații student(medie) din baza de date a facultatii, pe baza de nume și facultate(sau număr matricol)
            5. Actualizare liste medii
        ii. Administrator cămin
            1. Introducere informații căminul propriu
            2. Încheiere contract cu un student(adăugare student)
            3. Vizualizare liste, rapoarte
            4. Vizualizare situație locuri in caminul curent
            5. Ștergere student, caz lichidare/părăsire camin
        iii. Student
            1. Accesează informații cămin
            2. Introducere cerere de cazare – 3 opțiuni
            3. Completează documentele necesare și le încarcă online
            4. Confirmare/infirmare loc ocupat
            5. Vizualizare situație locuri în camin, caminul propriu
            6. Lichidare
            7. Posibilitate de realizare grupare
2) Aplicatie Desktop – cu acces la internet
   Este destinată folosirii in interiorul căminului, va fi instalată pe sistemele de calcul ale administratorilor din fiecare cămin.
    a. Utilizatori – drepturi
        i. Admnistrator
            1. Vizualizare lista studenți admisi la fiecare etapă:
                a. Afisare liste
                b. Liste după confirmări/contestații
        ii. Administrator cămin
            1. Vizualizare listă studenți admiși la fiecare etapa pentru căminul în cauză:
                a. Afisare liste

b. Liste după confirmări/contestații
c. Liste dupa grupări
2. Vizualizare cereri de grupare
3. Vizualizare situație cămin propriu
4. Generare rapoarte PDF

3) Aplicație pe telefon
   a. Utilizator:
      i. Student
         1. Reda posibilitatea de a primi diverse notificări pe parcursul procesului.
            Ex:
         - Vă informăm ca cererea dumneavoasta de cazare in caminu x a ajuns in atentia noastra.
         - Vă informăm ca ati fost repartizat in caminul x.
         - Vă rugăm sa intrati pe site pentru a va confirma locul in camin pana la data x.
         - Etc...
4) Baza de date realizeaza legatura dintre toate componentele de mai sus, punctul comun pe care toate se bazează.
5) Baza de date a facultatii – ofera acces doar administratorului sistemului pentru preluarea anumitor informatii.
6) Sistemul de autentificare si autorizare – necesar tuturor utilizatorilor primelor 3 componente amintite.
7) Algoritm de repartizare
   1) Preluare informații studenți
      a. Studentul face o cerere de cazare.
      b. Pe baza informatiior introduse datele despre student sunt preluate din baza de date a facultatii.
   2) Etape:
      a. Repartizare in functie de:
         i. prima opțiune
         ii. media studentului
         iii. numarul de locuri disponibile în fiecare camin
      b. Reactualizare:
         i. lista de studenti de repartizat
         ii. locuri ramase in fiecare camin
      c. primele doua etape se reiau pentru urmatoarele optiuni ale studentilor ramași
   3) Studentii sunt anuntati pe telefon de situatia curenta(admis/respins)
   4) Generare liste
   5) Unul din doi studenti admisi intr-un camin poate realiza o cerere de grupare:
   6) Logica grupare:
      a. Se verifica daca cei doi sunt admisi in acelasi camin
      b. Se calculeaza media gruparii
      c. Pe baza mediei, si camerelor disponibile se realizeaza repartizarea in camin
   7) Generare liste finale

- **prezentarea sistemului software din doua perspective (o diagrama pentru fiecare perspectiva + explicatii)**

1) Prima perspectivă



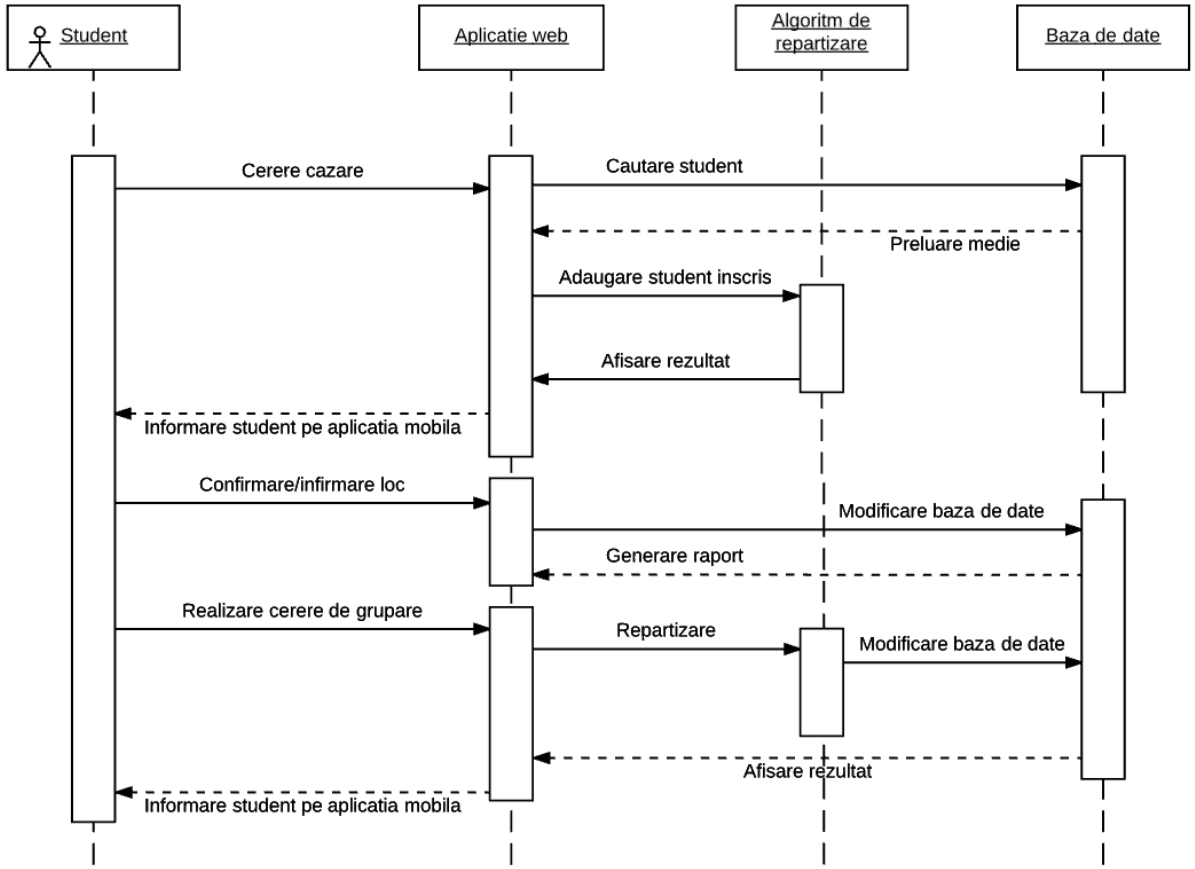
Figură 1. Diagrama de componenete

Administratorul de cămin, studentul, şi administratorul de sistem se pot autentifica/înregistra prin intermediul serverului de autentificare şi autorizare. În urma autentificării fiecare din cei trei utilizatori primeşte anumite drepturi de acces în aplicaţie în funcţie de tipul de utilizator.

Serverul cu baza de date a aplicaţiei comunica cu toate cele trei aplicaţii(Desktop, Web, Mobilă). Schimbul de informaţii cu aplicaţia mobilă şi desktop se realizează într-un singur sens.

Aplicaţia Web comunica şi cu serverul cu baza de date a facultăţii din care preia informaţii legate de mediile studenţilor.

2) A doua perspectivă



Figură 2. Diagrama de secvență

În primă fază, studentul iniţiază o cerere pentru cazare, completând formularele cerute în aplicaţia web. După ce media studentului este preluată, studenţii sunt supuşi unui algoritm de repartizare. Rezultatul acestuia este transmis către student la finalul procesului.

Următoarea etapă presupune o sesiune de confirmări în urma căreia studentul este adăugat în baza de date a căminului ales(în caz de confirmare).

La iniţiativa studentului, acesta poate opta pentru gruparea în cameră cu un alt student, realizând o cerere tip. După repartizare, baza de date e actualizată, iar studentul e informat.

- identitifcarea celor mai importanți 3 indicatori de calitate, specificarea masurii alese pentru fiecare indicator de calitate si argumetarea alegerii

1) *Scalabilitatea din punct de vedere al cresterii numarului de cereri simultane*
   Întrucât sistemul se poate extinde la peste 30 de cămine studenţeşti, numărul de conexiuni iniţiate în special în perioadele în care startul înscrierilor este dat, poate atinge cote maxime, iar blocarea sistemului la volum mare de date nu este permisă. Suplimentarea puterii de calcul se va realiza prin adăugarea mai multor procesoare şi a unei mai mari memorii.
2) *Securitatea informatiei transferate*
   Întrucât întregul proces are la bază un sistem de autentificare este necesar ca aplicația să verifice identitatea utilizatorilor și a celorlalte aplicații cu care comunică.

Odata cu autentificarea, utilizatorii primesc anumite drepturi de acces la resursele sistemului. Este necesar ca schimbul de mesaje sa fie criptat, iar documentele transmise sa fie semnate digital pentru a asigura integritatea si nerepudierea. Se va folosi tehnologia SSL/TLS.

3) *Integritatea*

Integrarea la nivelul datelor se poate realiza prin stocarea și manipularea datelor în așa fel încât alte aplicații să le poată accesa. Se vor dezvolta interfete API prin intermediul cărora să se poată accesa datele, în acest fel putând fi respectate anumite reguli(pentru un anumit set de date de intrare se vor primi doar anumite iesiri) și de asemenea, in acest mod, aplicația se poate extinde în viitor realizând diverse alte API-uri.

- **identificarea tehnologiilor middleware folosite pentru a comunica intre componente, argumentarea alegerilor**

Se va folosi modelul bazat pe cozi de mesaje deoarece in cazul in care cererile unor clienti(aplicatiile web si desktop) nu pot fi procesate imediat de catre server, sistemul sa nu se blocheze si sa permita in continuare utilizarea interfetelor.

De asemenea, serverul poate gestiona mai mulți clienți conectați simultan.

Astfel pentru a se realiza decuplarea expeditorului și a destinatarului sunt folosite una sau mai multe cozi în care expeditorii pot pune mesaje, respectiv din care destinatari pot citi mesaje. Un astfel de server poate să gestioneze mai multe cozi simultan, respectiv mai multe mesaje transmise simultan prin intermediul firelor de execuție organizate in pool-uri de fire de execuție. Un singur proces poate trimite mesaje către mai multe cozi, respectiv fiecare coadă poate fi interogată de unul sau mai mulți destinatari. Identificare cozilor se face pe baza numelor lor.

- **identificarea principalelor modele și stiluri arhitecturale folosite, argumentarea alegerilor**

Se va utiliza modelul client-server. Aplicaţiile(web, desktop, mobilă) vor fi clienţii care vor avea acces prin intermediul serverului la baza de date a sistemului.

Clienţii şi serverele vor comunica printr-o reţea de calculatoare(prin Internet). Clienţii nu partajează niciuna dintre resursele proprii, ci apelează la resursele serverului prin funcţiile server, inițiând sesiuni.

Se va utiliza MVC(Model-View-Controller) deoarece separă funcţionalitatea specifică domeniului pentru care este dezvoltat sistemul software de interfaţă grafică a aplicaţiei, permiţând dezvoltarea, întreţinerea şi testare separată a celor două părţi. Poate fi folosit atât pentru sistemul software de tip aplicaţie desktop cât şi pentru cel de tip aplicaţie web.

- **prezentarea scenariilor de validare a arhitecturii**

Scenarii posibile:

1) Studentul completează cererea de cazare => validare prin: primirea confirmării/infirmării faptului că cererea a ajuns în atenţia administratorilor.

2) Studentul completează cererea de lichidare => validare prin: primirea confirmării/infirmării.

3) Algoritmul de repartizare => validare prin: primirea informaţiei referitoare la căminul în care a fost cazat.

4) Studentul nu confirmă locul în cămin => validare prin: primirea unei notificări.

5) Studentul completează cererea de grupare => validare prin: primirea confirmării/infirmării faptului că cererea a ajuns în atenţia administratorilor.

6) Mai mulţi studenţi se înscriu simultan => validare prin: gestionarea corectă prin cozile de mesaje, ne asigurăm că informaţiile lor nu se pierd.

MOGA DELIA

An IV,
IS, IP,
Gr.2.1