

Cazare - TUIASI

Aplicație pentru gestionarea cazărilor în căminele din campusul
universitar “Tudor Vladimirescu” din Iași

Arhitectura sistemului

Nume proiect	Cazare - TUIASI
Nume document	Arhitectura sistemului / Architectural Design Document
Autori	Bîzgan Alina, Cojocaru Constantin-Cosmin, Mihalache Nicoleta-Ecaterina, Modiga Camelia-Maria, Moroșan Carla-Adriana
Reviewed by	
Status	Terminat

Versiuni

Data	Versiune	Modificări
20.10.2021	v0.1	Structura documentului
27.10.2021	v0.2	Actualizare document
20.11.2021	v1.0	Versiune stabilă document

4.01.2022

v1.2

Actualizare design servicii și schemă bază de date

Cuprins

Introducere	3
1.1 Scopul documentului	3
1.2 Scurtă descriere a proiectului	3
1.3 Definiții și abrevieri	3
2 Prezentarea sistemului	3
2.1 Tehnologii folosite	4
2.1.1 Frontend	4
2.1.2 Backend	4
2.1.3 Deployment	5
3 Arhitectura top-level	5
3.1 Identificarea componentelor	5
3.1.1 Server-ul de servicii	6
3.2 Relațiile și comunicarea dintre componente	7
3.1.2 Server-ul de baze de date	8
4 Descrierea componentelor	9
4.1 Frontend	9
4.2 Backend	9
4.1.1 Baza de date	9
4.1.5 REST API Student Management	10
4.1.7 UAC + Account Management	11
4.1.8 REST API Accommodations Management	12
4.1.9 API Gateway	12
5 Concluzii	12

1 Introducere

1.1 Scopul documentului

Documentul “Definirea cerințelor” pune la dispoziție o descriere amănunțită a arhitecturii sistemului aplicației pentru gestionarea cazărilor în căminele din campusul studentesc “Tudor Vladimirescu” din Iași.

Acest document pune la dispoziție o descriere amănunțită a sistemului.

1.2 Scurtă descriere a proiectului

Proiectul **Cazare - TUIASI** este o platforma pentru gestionarea eficientă a procesului de cazare în toate cele 21 de cămine situate în Campusul “Tudor Vladimirescu” din Iași. Platforma online are scopul de a ușura munca administrației căminelor prin oferirea unei desfășurări simple și rapide, evitând aglomerația, reducând timpul de așteptare, dar și de a elimina eventualele erori care pot apărea în procesul de cazare. De asemenea aplicația poate fi folosită de către studenți pentru alegerea camerei de cămin, achitarea taxei de cazare.

1.3 Definiții și abrevieri

- TUIASI – Universitatea Tehnică “Gheorghe Asachi” Iași

2 Prezentarea sistemului

Cazare- TUIASI este proiectat după arhitectura client-server. Aceasta arhitectura constă în comunicarea la distanță între mai multe componente printr-un protocol de comunicație web, prin trimiterea de cereri pentru obținerea resurselor și primirea de răspunsuri cu resursele aferente.

Componentele principale care formează structura platformei sunt:

- Interfața cu utilizatorul: fereastra de comunicare cu utilizatorul.
- Server-ul web: este componenta de procesare a cererilor trimise de utilizator în urmă navigării pe interfață.
- Server-ul de servicii web: aceasta componenta are rolul de a prelua datele stocate de către utilizator, de a prelucra datele preluate de la server-ul web sau din baza de date și de a trimite aceste prelucrări sub forma de răspunsuri către server-ul web.
- Server-ul de baza de date: componenta are funcționalitatea de a stoca date pentru a fi preluate ulterior la nevoie.



Fig. 1 Privirea de ansamblu al sistemului cu componentele constitutive principale

2.1 Tehnologii folosite

2.1.1 Frontend

Interfața cu care interacționează utilizatorul se realizează folosind limbajul de programare Javascript, împreună cu limbajul de marcare HTML și CSS. Pentru interfața web se folosește și framework-ul Angular. Angular este unul dintre cele mai populare tehnologii de dezvoltare front-end, alături de React și Vue.js. Este bazat pe TypeScript (superset sintactic al limbajului JavaScript) și oferă toate avantajele generale ale limbajelor de programare.

2.1.2 Backend

Pentru implementarea server-ului web se utilizează Angular dev server. Aceasta facilitează managerierea paginilor web, trimiterea și primirea de cereri prin capabilitățile puse la dispoziție de server-ului web.

Tehnologiile folosite pentru implementarea server-ului de servicii web sunt: Java, framework-ul Spring și facilitățile oferite de acesta. Java este un limbaj de programare orientat obiect cu care se pot dezvolta atât aplicații desktop, cât și web. SpringBoot este un framework destinat dezvoltării de aplicații web oferind modalități ușoare de integrare cu alte module, acesta fiind unul dintre motivele pentru care am ales acest framework.

Tipul de bază de date pe care o va folosi platforma este MySQL. Baza de date este indispensabilă pentru dezvoltarea aplicațiilor web. Un exemplu de utilizare a bazei de date este acela că majoritatea aplicațiilor necesită autentificarea utilizatorilor pentru a stabili diferite roluri și servicii pentru aceștia. Astfel ca utilizatorii trebuie să aibă cont valid, iar acesta nu poate fi stocat decât într-o bază de date.

Arhitectura serviciilor web REST

Serviciile web REST sunt servicii slab cuplate, potrivite pentru crearea de API-uri destinate clienților din Internet. Acronimul REST provine de la „REpresentational State Transfer”, un stil

arhitectural de creare a aplicațiilor client-server orientate pe transferul reprezentărilor de resurse prin cereri și răspunsuri.

Aceste servicii se bazează pe comunicațiile prin protocolul fără stare HTTP (Hypertext Transfer Protocol - RFC 7231). Deoarece stilul arhitectural REST este orientat pe resursă, atunci comunicațiile reprezintă practic operațiile care se pot efectua asupra resurselor puse la dispoziție.

Model-View-Controller (MVC)

MVC este un șablon arhitectural folosit pentru organizarea codului. Acesta este backbone-ul principal al serviciilor web. Șablonul definește 3 componente decuplate: View-ul: componenta responsabilă de serializare/deserializarea resurselor, Controller-ul: componenta responsabilă pentru primirea și trimiterea cererilor și Modelul: componenta responsabilă pentru persistența și preluarea datelor din baza de date. Modelul arhitectural MVC este o alegere optimă pentru a oferi scalabilitate și reutilizabilitate la nivel de serviciu web.

2.1.3 Deployment

Deploymentul întregii platforme este realizat utilizând platforma de containerizare Docker. Fiecare serviciu web este încapsulat într-un container care va rula pe un proces separat. Fiecare container expune un port prin intermediul căruia se poate accesa serviciul respectiv.

3 Arhitectura top-level

3.1 Identificarea componentelor

Acest capitol urmărește descompunerea fiecărei părți al sistemului descris în capitolul anterior.

Respectând stilul arhitectural client-server, arhitectura platformei urmărește să ofere o decuplare cât mai mare între componentele principale: Frontend și Backend. Partea de back-end manageriază accesul utilizatorilor la resursele sistemului și stocarea datelor. Resursele constituite părții de frontend sunt paginile web pe care navighează utilizatorul.

În cazul platformei **Cazare - TUIASI**, structura backend-ului este alcătuită din 3 componente:

- Server-ul web
- Server-ul de servicii web
- Server-ul de baze de date

În cazul platformei **Cazare - TUIASI**, structura frontend-ului este alcătuită din următoarele pagini web:

- Pagină login
- Pagină înregistrare cont nou (admin)
- Pagina înregistrare student (formular cerere cazare)
- Pagina dashboard Admin
- Pagina dashboard Student
- Pagina unauthorized access pe care utilizatorul poate ajunge în cazul în care accesează o pagină pentru care nu are drepturi de acces

3.1.1 Server-ul de servicii

Server-ul de servicii web reprezintă nucleul de procesare a datelor în sistemul de fază. Server-ul de servicii web urmărește transpunerea cerințelor funcționale a aplicației din documentul URD în componente individuale cu responsabilități bine definite. O astfel de abordare în design-ul server-ului de servicii web crește scalabilitatea și permite reutilizarea componentelor care îl compun.

- UAC service
 - Este punctul inițial de intrare în aplicație. Pe baza contului cu care se accesează, această componentă va genera un token de tip JWT, care va permite accesul la resurse pe baza permisiunilor specifice tipului de utilizator autentificat. Acesta conține și endpoint-urile care tratează cererile care manageriază conturile utilizatorilor.
- API Gateway
 - Conține toate endpoint-urile cunoscute user-ului, accesibile cu un token de tip JWT. Va redirecta fiecare cerere către serviciul responsabil, care se va ocupa la rândul lui de cererea în cauză.
- Room repartition service
 - Tratează logica de organizare (update/populare) a tabelor din baza de date ce țin de camere/cămine și studenți, pentru a asigura repartiția acestora în etapa de cazare.
- Student management service
 - Conține endpoint-urile care tratează cererile referitoare la informațiile personale ale studenților, tratează logica de repartiție a studenților în baza preferințelor alese. Cererile urmează paradigma CRUD: adăugare, actualizare, ștergere etc.
- Accommodations management service

- Conține endpoint-urile care tratează cererile de manageriere a căminelor și camerelor din cămine. Cererile permit doar preluarea de date deoarece modificarea acestora se întâmplă doar în cazuri excepționale.

3.2 Relațiile și comunicarea dintre componente

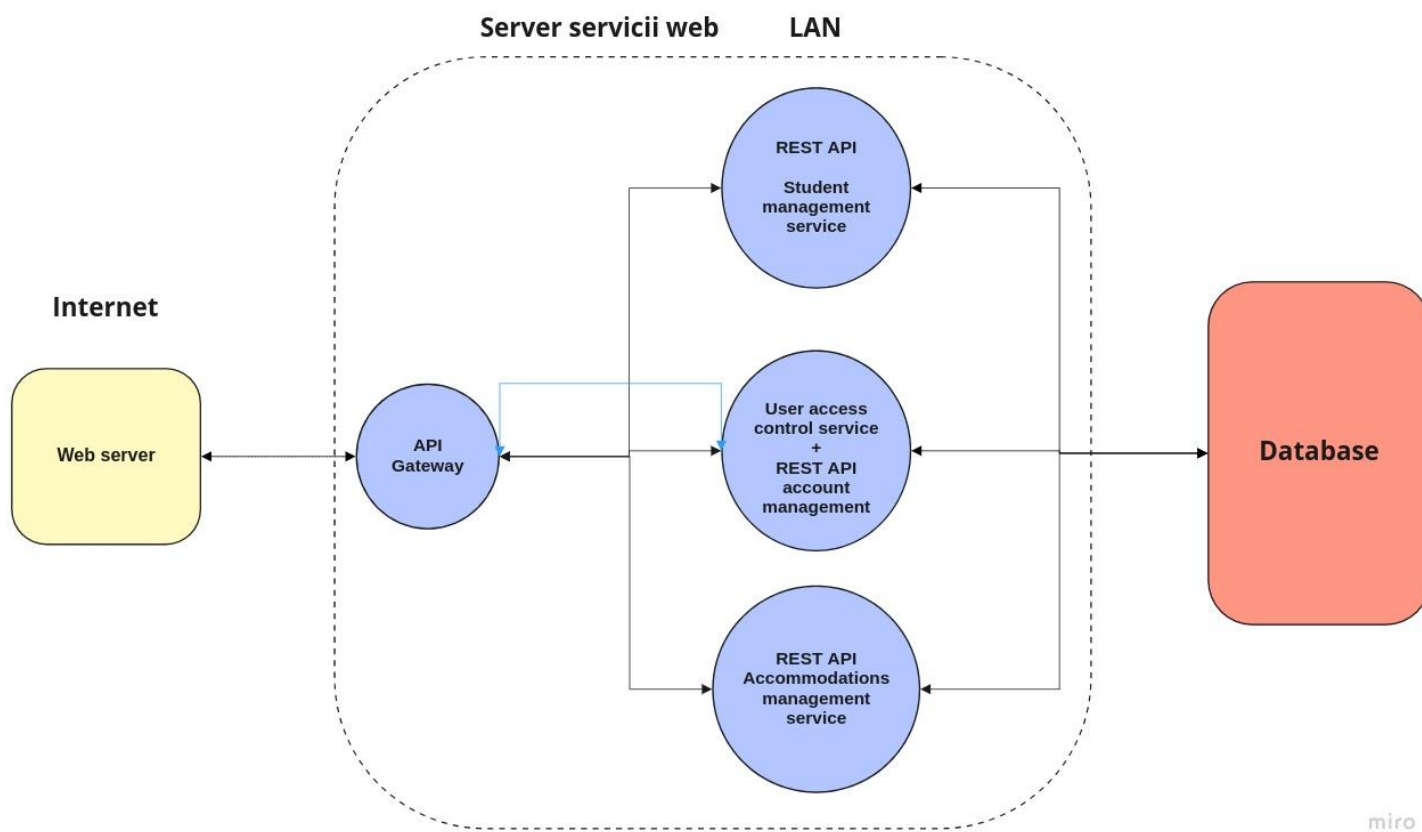


Fig. 2 Împărțirea pe servicii a server-ului de servicii web

3.1.2 Server-ul de baze de date

Platforma **Cazare - TUIASI** utilizează un singur server de baze de date. Server-ul de baze de date cu care se interfațează platforma este MySQL. Platforma utilizează o singură bază de date în care sunt stocate toate tabelele folosite de aceasta. Schema bazei de date **accommodations** este următoarea:

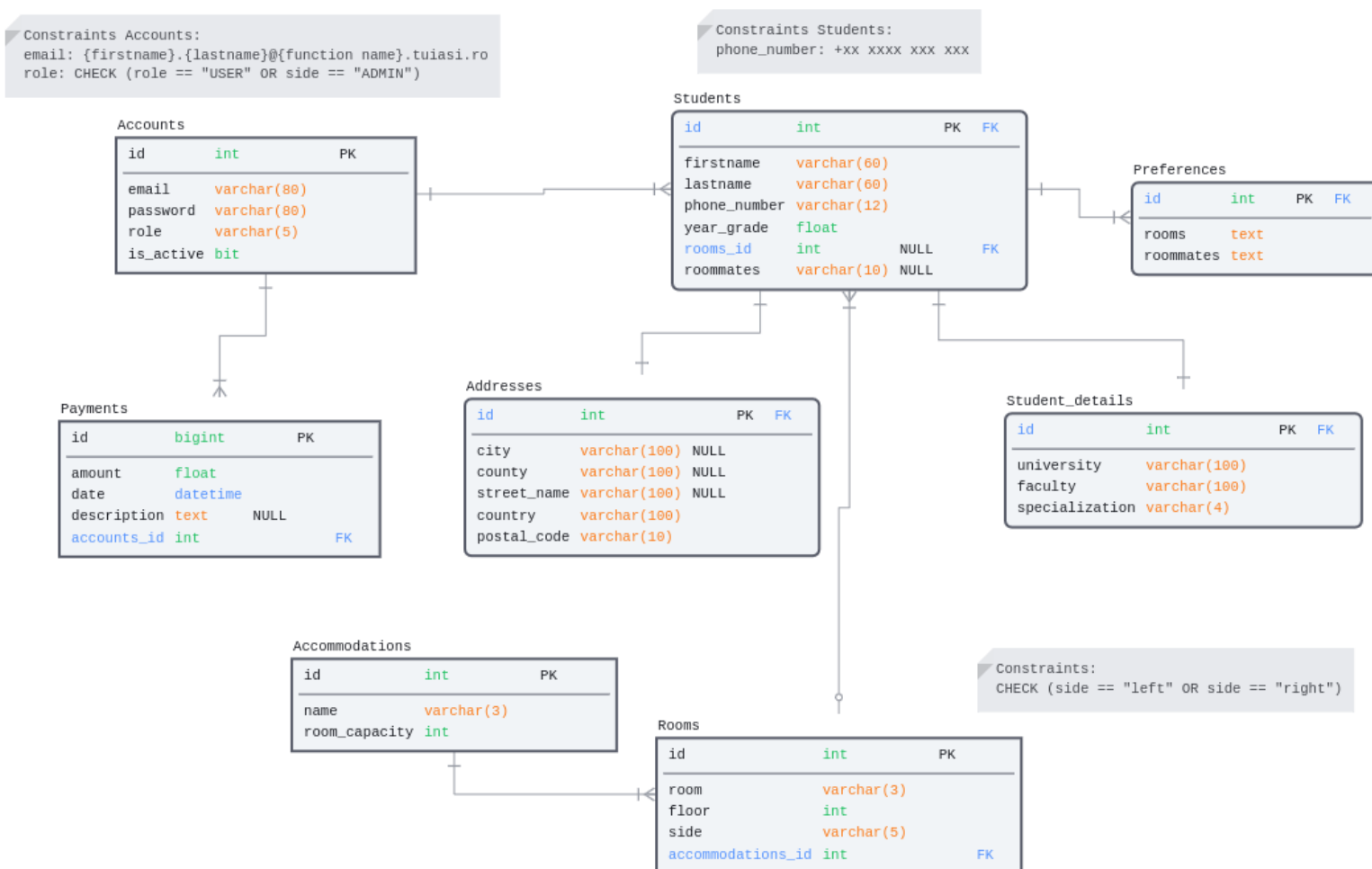


Fig. 3 Schema bazei de date **accommodations**

4 Descrierea componentelor

4.1 Frontend

- Pagina cerere cazare
- Pagina autentificare
- Pagina status repartiție
- Pagina dashboard student
- Pagina dashboard administrator
- Pagina acces neautorizat

4.2 Backend

4.1.1 Baza de date

- Baza de date este o componentă independentă care stochează informațiile legate de: studenți, cămine, camere, preferințe, situații legate de cazări etc.
- Această componentă nu depinde de alte componente.
- Baza de date poate este gestionată de către serviciile web.
- Aplicația folosește un Server MySQL pentru managerierea tabelelor din baza de date.
- Pentru fiecare tabel din baza de date se va folosi câte un repository care va implementa operațiile de bază cu acestea (INSERT, DELETE, SELECT, UPDATE).
- Din motive de securitate, baza de date este manageriata de către doi utilizatori cu permisiuni diferite.
 - Primul utilizator are doar permisiunea de a citi datele din orice tabelă.
 - Al doilea utilizator are permisiunea de a citi, adăuga, modifica și șterge datele din orice tabelă.

4.1.5 REST API Student Management

- Încapsulează funcționalitățile CRUD pentru managerierea preferințelor studenților.
- Stochează datele în tabelele corespunzătoare din baza de date.
- Această componentă preia datele completate de student din formularul de cazare.
- Încapsulează funcționalitățile CRUD pentru managerierea preferințelor studenților.
- Primește din interfață date referitoare la opțiunile studenților referitoare la cămin/cameră/colegi de cameră/anularea rezervării unei camere.
- Stochează aceste date în tabelele corespunzătoare din baza de date.

Endpoints:

Pentru managerierea datelor studenților

- **GET student-management-service/accommodations/students** - Pentru preluarea tuturor studenților înregistrați
- **GET student-management-service/accommodations/students/{id}** - Pentru preluarea datelor studentului cu id-ul **id**
- **GET student-management-service/accommodations/students/{id}/roommates** - Pentru preluarea colegilor de camera a studentului cu id-ul **id**
- **GET student-management-service/accommodations/students/accommodated** - Pentru preluarea tuturor studenților cazați
- **POST student-management-service/accommodations/students/{id}** - Pentru crearea studentului cu id-ul **id**
- **DELETE student-management-service/accommodations/students/{id}** - Pentru ștergerea studentului cu identificatorul unic **id**

Pentru managerierea adreselor studenților

- **GET student-management-service/accommodations/students-addresses** - Pentru preluarea adreselor tuturor studenților înregistrați
- **GET student-management-service/accommodations/students/{id}/addresses** - Pentru preluarea adresei pentru studentul cu id-ul **id**
- **POST student-management-service/accommodations/students/{id}/addresses** - Pentru crearea adresei pentru studentul cu id-ul **id**
- **DELETE student-management-service/accommodations/students/{id}/addresses** - Pentru ștergerea adresei pentru studentul cu id-ul **id**

Pentru managerierea datelor suplimentare despre studenți

- **GET student-management-service/accommodations/students-details** - Pentru preluarea detaliilor tuturor studenților
- **GET student-management-service/accommodations/students/students-details/{id}** - Pentru preluarea datelor suplimentare despre studentul cu id-ul **id**
- **POST student-management-service/accommodations/students/{id}/students-details** - Pentru adaugarea datelor suplimentare despre studentul cu id-ul **id**
- **DELETE student-management-service/accommodations/students-details/{id}** - Pentru ștergerea datelor suplimentare despre studentul cu id-ul **id**

Pentru managerierea preferințelor studenților

- **GET preferences-management-service/accommodations/students/{id}/preferences** - Pentru preluarea preferinței pentru studentul id-ul **id**
- **GET preferences-management-service/accommodations/preferences/registration-status** - Pentru preluarea statusului de înregistrare. Acesta specifica dacă încă se admit cereri de cazare.
- **POST preferences-management-service/accommodations/students/{id}/preferences** - Pentru crearea preferinței pentru studentul cu id-ul **id**
- **POST preferences-management-service/accommodations/preferences/start-repartition** - Pornește procesul de repartiție a studenților.
- **DELETE preferences-management-service/accommodations/students/{id}/preferences** - Pentru ștergerea preferinței pentru studentul cu id-ul **id**

4.1.7 UAC + Account Management

- Autorizează accesul utilizatorilor pentru a folosi sistemul.
- Întoarce token-uri pentru accesarea resurselor din spatele porții API.
- Definește permisiunile de acces la resurse.
- Componentă responsabilă cu managerierea conturilor de utilizator din platformă. Acestea vor fi stocate ulterior în tabela corespunzătoare din baza de date.
- Pe baza contului nou creat user-ul se poate autentifica. Componenta verifică credențialele introduse și oferă acces în aplicație.
- Permite efectuarea operațiilor de schimbare asupra datelor contului de student: primește noile date din interfață și actualizează tabelele corespunzătoare din baza de date.

Endpoints:

- **POST uac/authenticate** - Pentru validarea datelor unei persoane și trimiterea ca răspuns al unui JWT
- **POST uac/validate-token**
- **GET uac/accommodation/accounts** - Pentru preluarea tuturor conturilor
- **GET uac/accommodation/accounts/{id}** - Pentru preluarea unui cont cu id-ul **id**
- **PUT uac/accommodation/accounts/{id}** - Pentru modificarea contului
- **POST uac/accommodation/accounts** - Pentru crearea contului
- **DELETE uac/accommodation/accounts/{id}** - Pentru ștergerea contului cu id-ul **id**

4.1.8 REST API Accommodations Management

- Încapsulează funcționalitățile CRUD pentru managerierea camerelor și a căminelor.
- Stochează datele în tabelele corespunzătoare din baza de date.

Endpoints:

- **GET accommodations-service/accommodations/accommodations** - Pentru preluarea tuturor căminelor.
- **GET accommodations-service/accommodations/rooms** - Pentru preluarea tuturor camerelor.
- **GET accommodations-service/accommodations/rooms/{accommodationId}** - Pentru preluarea unei camere al cărei cămin are id-ul **id**.

4.1.9 API Gateway

- Fiecare cerere primită din interfață va ajunge în API Gateway care va redirecta cererea către componenta responsabilă.

5 Concluzii

Platforma online **Cazare - TUIASI** are scopul de a ușura procesul de cazare în căminele din campusul studentesc “Tudor Vladimirescu” din Iași prin oferirea unei variante mai simple de a gestiona repartizarea studenților în cămine. Astfel, prin intermediul aplicației, administratorii căminelor au o situație clară a fiecărui student, iar studenții își pot alege cu ușurință un loc în cămin.