

Architectural

Design

Document

StudentStay Project

Identificarea proiectului

Nume proiect	StudentStay
Nume document	Arhitectura sistemului/ADD
Echipa	WebMinds
Status	Done

Cuprins

1. [Introducere](#)
 - 1.1. Scopul documentului
 - 1.2. Scurtă descriere a proiectului
2. [Arhitectura sistemului](#)
 - 2.1 Descriere generală
 - 2.2 Tehnologii cheie
 - 2.3 Straturile aplicației
 - 2.4 Diagrama de arhitectură
3. [Tehnologii](#)
4. [Analiza Fluxului de Funcționare](#)
5. [Microservicii](#)
6. [Aplicație desktop](#)
7. [Concluzie](#)

1. Introducere

1.1. Scopul documentului

Acest document oferă o descriere detaliată a arhitecturii propuse pentru aplicația StudentStay, destinată gestionării camerelor de cămin la Universitatea Tehnică „Gheorghe Asachi” din Iași. Documentul include design-ul arhitectural al aplicației, fluxurile de date și tehnologiile utilizate. Descrie arhitectura sistemului, componentele majore, interacțiunea acestora și infrastructura necesară pentru implementarea și rularea aplicației. De asemenea, include informații despre performanță, securitate și tehnologiile software utilizate.

1.2. Scurtă descriere a proiectului

Aplicația facilitează gestionarea camerelor de cămin pentru studenții Universității TUIASI. Funcționalitățile principale includ: trimiterea și verificarea documentelor, rezervarea camerelor în funcție de medie și alte criterii, gestionarea clasamentului studenților și integrarea unui sistem de notificări.

2. Arhitectura sistemului

2.1. Descriere generală

Sistemul StudentStay utilizează o arhitectură bazată pe microservicii, în care fiecare serviciu gestionează o funcționalitate specifică (ex: autentificare, alocarea camerelor, gestionarea documentelor, etc). Această arhitectură modulară permite scalarea aplicației în funcție de numărul de cereri.

2.2. Tehnologii cheie

Pentru dezvoltarea proiectului vom utiliza:

- Angular: server frontend;
- Spring Boot: microservicii backend
- Amazon Textract: extragerea textului și catalogarea documentelor
- Amazon S3: stocarea documentelor/fișierelor
- Google OAuth 2.0: autentificare prin contul instituțional

- PostgreSQL: pentru salvarea datelor utile fiecărui serviciu
- Amazon SNS (Simple Notification Service): pentru notificări

2.3. Straturile aplicației

Aplicația este împărțită în 3 straturi principale:

2.3.1. Presentation Layer (Stratul de prezentare, Angular)

- Reprezintă partea vizibilă, interactivă a platformei. Oferă interfața pentru studenți, administrator și membrii comisiei.
- Interacționează cu serviciile de backend prin RESTful APIs.

2.3.2. Service Layer (Microservicii Spring Boot)

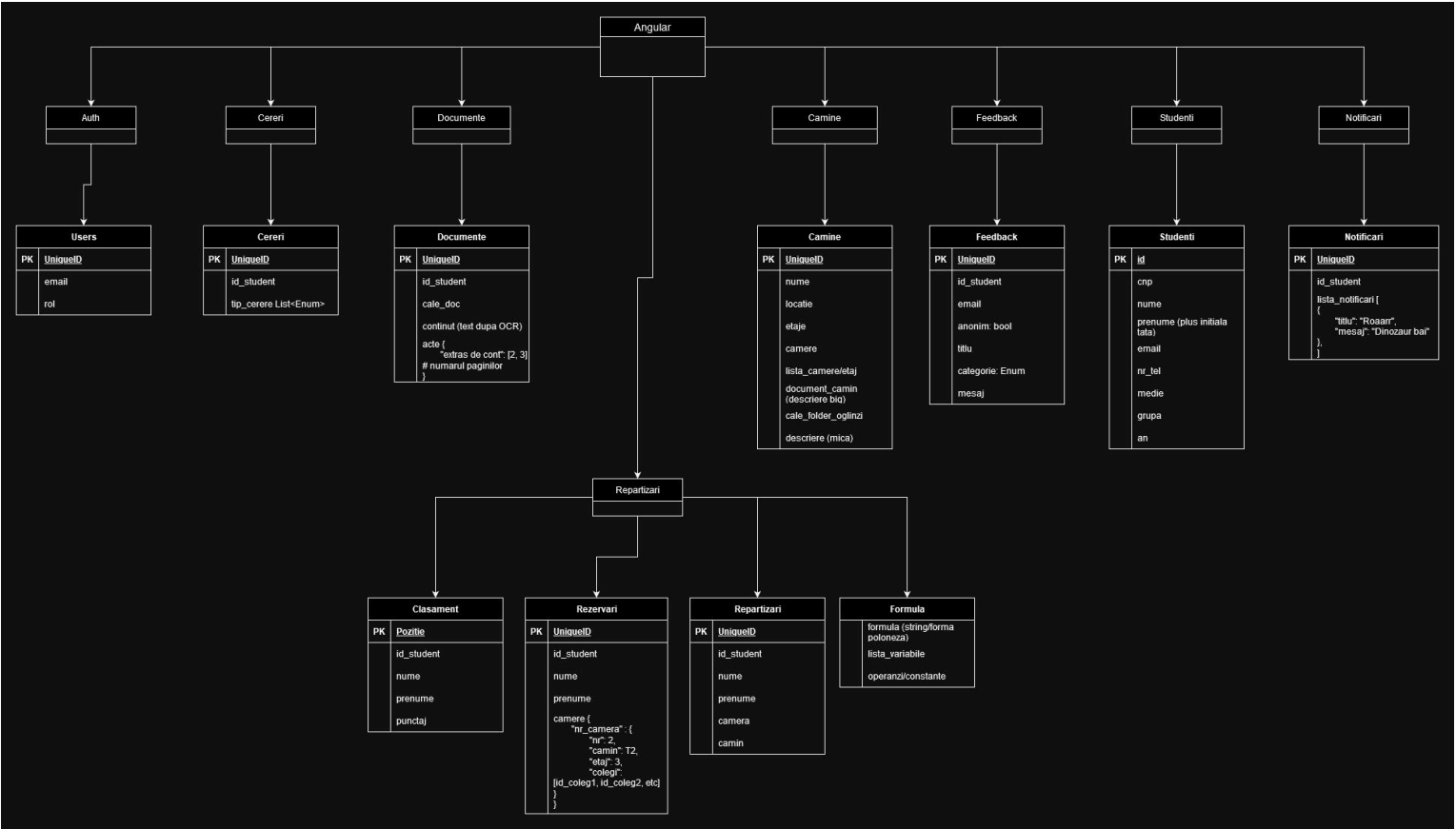
- Manageriază logica business-ului, API endpoints si cererile.
- Respectă principiile SOLID, asigurându-se că fiecare serviciu are o singură responsabilitate.
- Format din următoarele microservicii:

- Autentificare
- Cereri
- Documente
- Cămine
- Repartizări
- Notificări
- Feedback
- Studenți

2.3.3. Data Layer (PostgreSQL)

- Stochează datele utilizatorilor, disponibilitatea camerelor, cererile, etc.
- Permite stocarea datelor personalizate, precum obiecte JSON

2.4. Diagrama de arhitectură



3. Tehnologii

3.1. Angular (Front-End):

Angular este un framework open-source JavaScript, dezvoltat de Google, utilizat pentru dezvoltarea front-end-ului aplicațiilor web. În cadrul aplicației noastre, vom crea interfețele utilizatorilor, atât pentru studenți, cât și pentru membrii comisiei, folosind setul de instrumente pus la dispoziție de biblioteca de componente UI Angular Material. Aceasta conține o colecție de butoane, casete, meniuri sau alte utilitare pentru gestionarea layout-ului. Comunicarea cu microserviciile din backend se va realiza prin intermediul API-urilor REST, acestea gestionând datele între interfața utilizatorului și server. Atunci când un user declanșează o acțiune (ex. trimite formular, încarcă date etc.), Angular trimite cereri HTTP către serviciile aferente prin intermediul REST. Fiecare microserviciu va procesa aceste cereri, interacționând cu baza de date sau alte resurse necesare și va răspunde cu datele cerute.

3.2. Spring Boot (Back-End):

Spring Boot reprezintă un framework Java bazat pe Spring care simplifică dezvoltarea de aplicații web și microservicii. Acesta permite construirea de aplicații autonome cu un set minim de configurații. Pentru comunicarea cu front-end-ul, Spring Boot folosește API-uri RESTful. Atunci când clientul trimite o cerere HTTP, framework-ul va procesa cererea și va interacționa cu baza de date (dacă este cazul) pentru a obține informațiile necesare. După procesare, Spring Boot va trimite un răspuns, de obicei în format JSON/XML către aplicația front-end sau alte microservicii. De asemenea, backend-ul va avea funcționalități integrate pentru securitate, configurarea de baze de date, gestionarea sesiunilor sau a tranzacțiilor. În continuare, Spring Boot permite gestionarea cererilor de cazare, verificarea documentelor, actualizarea informațiilor sau comunicarea între studenți, prin definirea de endpoint-uri pentru operații de creare, citire, actualizare sau ștergere.

3.3. Amazon Textract:

Amazon OCR (Optical Character Recognition) este un serviciu care utilizează tehnici de machine learning pentru a extrage textul din imaginile și documentele scanate din cadrul aplicației. Acesta va fi utilizat pentru analiza documentelor încărcate de studenți pe platformă (de ex. buletine, diverse certificate pentru validarea priorității/gratuității), astfel reușind să alocăm fiecărui student camera corespunzătoare. Pentru a ne asigura că fiecare student a încărcat documentele necesare, vom prelua informații precum numele, dar și o listă de cuvinte cheie care vor ajuta membrii comisiei de cazare să identifice dosarele complete (iar în cazul în care nu sunt complete, să semnaleze acest lucru). Astfel, procesul de încărcare și validare al documentelor va fi complet automatizat, simplificând procedura pentru studenți, fără a fi necesară intervenția manuală a persoanelor autorizate din comisie.

3.4. PostgreSQL:

Pentru partea de persistență a datelor, vom utiliza baze de date relaționale. În cadrul aplicației noastre, vom folosi PostgreSQL în stocarea informațiilor relative la cererile de cazare, autentificare, documente, repartizări, cămine, feedback, notificări respectiv studenți. Pentru fiecare categorie (serviciu) menționată vom avea o bază de date cu tabelele aferente. SGBD-ul în cauză este deseori asociat cu noțiunea de versatilitate pentru capacitatea de stocarea a datelor particularizare (obiecte custom).

3.5. Amazon S3 (Simple Storage Service):

Pentru a stoca documentele corespunzătoare fiecărui student, S3 bucket permite gestionarea de fișiere și foldere stocate în cloud. Prin intermediul acestui API, aplicația poate crea, citi, actualiza sau șterge documente, facilitând astfel stocarea și partajarea eficientă. Astfel, utilizatorii de tip membru comisie cazare vor putea accesa în timp real documentele, indiferent de locație, în mod

scalabil și flexibil. Acest serviciu, combinat cu cel de identificare a textului din imagine, vor contribui la un proces de cazare eficient pentru studenți.

3.6. OAuth 2.0:

OAuth 2.0 este un protocol de autorizare care permite aplicației să acceseze resursele utilizatorilor fără a utiliza parola propriu-zisă. Acesta funcționează prin oferirea unui token de acces pe care aplicația îl folosește în interacțiunea cu API-urile, asigurându-se astfel securitatea și protecția datelor fiecărui utilizator. În cadrul aplicației StudentStay, orice user trebuie să aibă valid un email instituțional (@tuiasi.ro), orice alt domeniu fiind restricționat. După autentificare, fiecare utilizator va avea acces la funcționalitățile specifice, în funcție de rolul său.

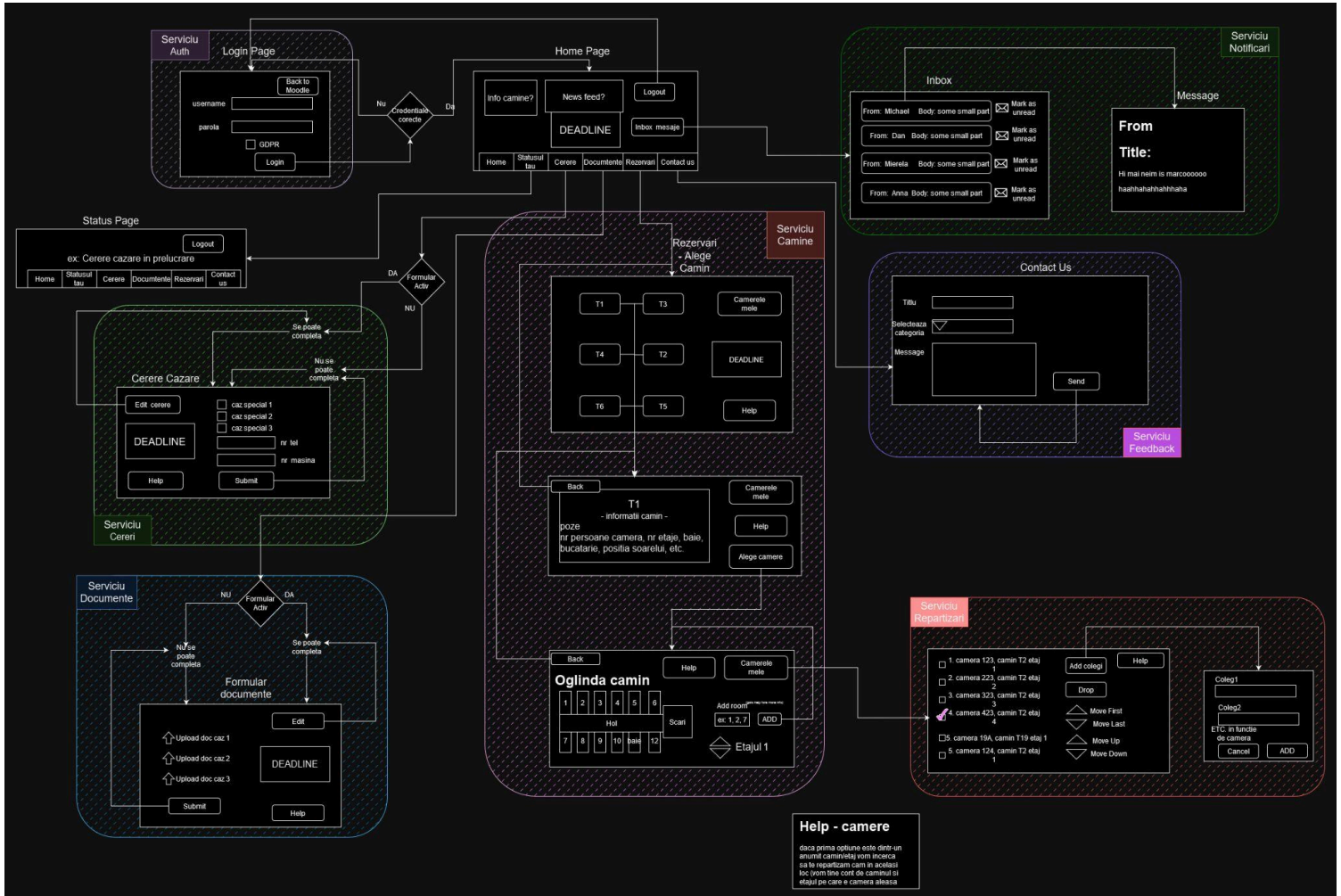
4. Analiza Fluxului de Funcționare

4.1. Fluxul aplicației din perspectiva studentului

Această diagramă ilustrează interacțiunea principală a unui student în cadrul aplicației pentru cazarea la cămin. Diagrama acoperă mai multe funcționalități esențiale:

- **Login și Autentificare:** Studentul începe procesul prin autentificarea în aplicație folosind un cont. După autentificare, este redirecționat către pagina principală, unde sunt disponibile diverse servicii.
- **Cerere de Cazare:** Studentul poate completa o cerere de cazare, aceasta este validată și procesată automat, studentul fiind informat despre starea cererii.
- **Documente:** Studentul va atașa documentele necesare.
- **Servicii de notificare și feedback.**

Architectural Design Document - ADD

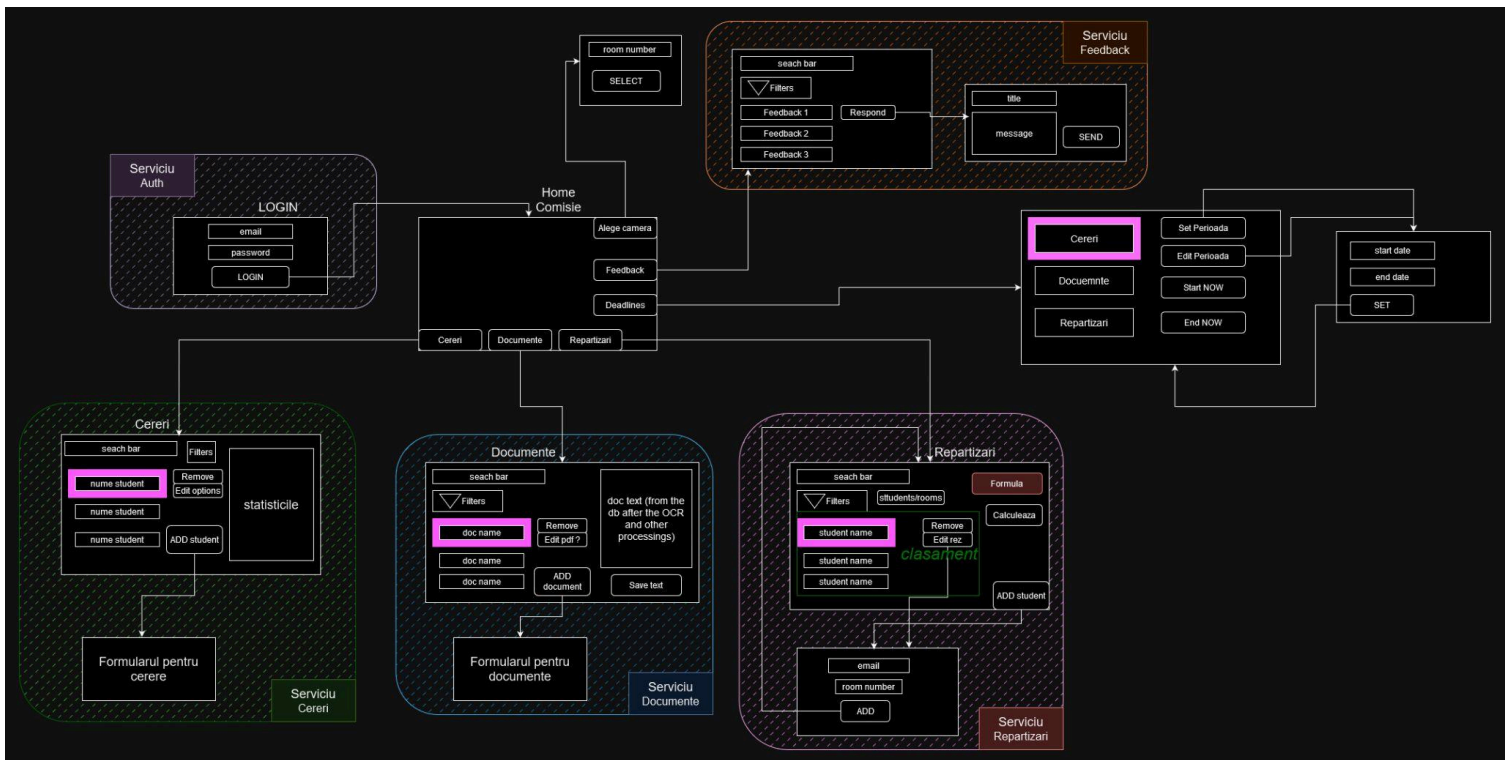


FlowChart - Student

4.2. Fluxul aplicației din perspectiva membrului comisiei

Diagrama reprezintă procesul prin care comisia de cazare interacționează cu cererile și documentele studenților.

- **Gestionarea cererilor:** Comisia are acces la toate cererile depuse de studenți, pe care le poate filtra, edita sau procesa.
- **Gestionarea documentelor:** Comisia are acces la toate documentele depuse de studenți, pe care le poate filtra, edita sau procesa.
- **Repartizarea studenților în cămine:** Comisia poate selecta și alocă camere în funcție de cerințele fiecărui student, iar această repartizare este validată automat.
- **Feedback:** Există un serviciu dedicat prin care comisia poate primi feedback și poate răspunde la sugestiile sau problemele întâlnite de studenți.

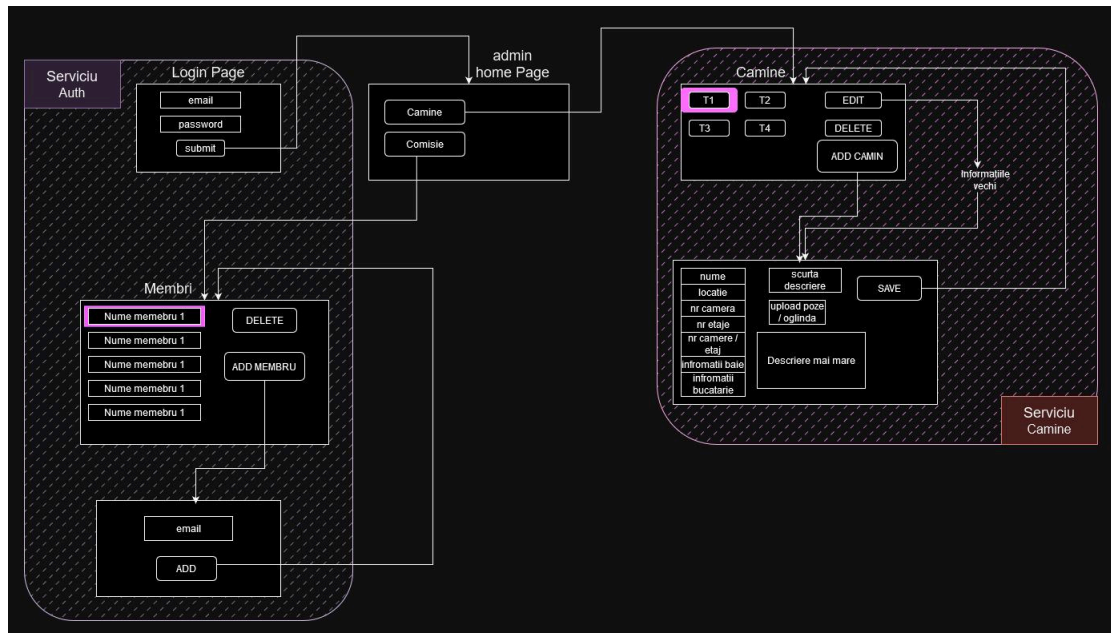


FlowChart - Comisie

4.3. Fluxul aplicației din perspectiva administratorului

Această diagramă descrie funcționalitățile disponibile pentru administratori în cadrul aplicației, precum:

- **Gestionarea conturilor de utilizatori:** Administratorii pot crea, edita sau șterge membrii comisiei sau studenți, și le pot atribui roluri specifice în funcție de nevoile sistemului.
- **Gestionarea informațiilor despre cămine:** Administrația poate adăuga, edita sau șterge informațiile despre cămine (de exemplu, număr de camere, locație, facilități).
- **Autentificare și navigare:** La fel ca și utilizatorii obișnuiți, administratorii se autentifică și navighează printr-un sistem de meniuri simplu, prin care pot accesa funcționalitățile de administrare.



FlowChart - Admin

5. Microservicii

5.1. Autentificare

Fluxul autentificării unui utilizator este manageriat de serviciul Autentificare ce utilizează OAuth pentru logarea utilizatorilor prin intermediul credențialelor instituționale. Serviciul generează tokenuri de acces securizate pentru interacțiunile ulterioare cu serviciile de backend.

Tehnologii: Spring Security, OAuth 2.0

Flux (/auth):

- 1.Utilizatorul inițializează logarea prin intermediul front-end-ului .
- 2.Serviciul de autentificare validează credențialele folosind OAuth.

5.2. Cereri

Acest serviciu procesează cererile de cazare venite de la studenți.

Tehnologii: Spring Boot

Flux (/cereri):

1. Studentul încarcă trimite cererile. (/cereri/cerere-nouă)
2. Studentul editează cererea. (/cereri/editează)

5.3. Documente

Rolul acestui serviciu este de a încărca și vizualiza documentele trimise de studenți. Pentru a prelua elemente text care ajută la identificarea persoanei, respectiv a tipului de prioritate în care se încadrează, serviciul integrează Amazon OCR. Astfel, conținutul fișierelor este extras automat, iar verificarea manuală de către membrii comisiei nu mai este necesară. Pentru a stoca aceste documente în cloud, vom utiliza Amazon S3, acest API facilitând accesul la distanță pentru documentele fiecărui student.

Tehnologii: Amazon Textract, Amazon S3

Flux: (/documente)

1. Studentul încarcă documentele. (/documente/încarcă)

2. Serviciul trimite documentele către Amazon OCR pentru extragerea cuvintelor cheie.
3. Cuvintele cheie sunt validate în funcție de criteriile pentru care se aplică (tipul documentului identificat). (/documente/validare)
4. Dacă documentele se încadrează în cele specificate, vor fi stocate pe drive în spațiul corespunzător fiecărui student.

5.4. Repartizare

Serviciul de repartizare a camerelor manageriază alocarea fiecărui student a unei camere, utilizând preferințele selectate de aceștia (sub formă de opțiuni) și ierarhia deja calculată după formula impusă.

Tehnologii: Spring Boot

Flux: (/repartizări)

1. Studentul selectează o serie de opțiuni de camere, din cele existente în cămine (/repartizări/rezervă).
2. Este calculat punctajul fiecărui student și întocmită o ierarhie de repartizare (repartizări/clasament).
3. Se realizează automat repartizarea studenților pe camere (/get-room-allocation).

5.5. Cămine

Acest serviciu este apelat pentru realizarea operațiilor referitoare la datele căminelor.

Flux: (/cămine)

5.6. Notificări

În funcție de situație, studenții vor avea acces la notificări pentru dosare incomplete, deschiderea perioadei de selecție, etc. Acestea vor fi transmise prin

email și vor cuprinde informații precum deadline-uri, actualizări ale dosarelor etc.

Tehnologii: Spring Boot, Amazon SNS

Flux: (/notificări)

1. Crearea notificării - în funcție de situație, se generează notificarea specifică;
2. Transmitere notificare - prin email, la adresele instituționale ale studenților;

5.7 Feedback

Întrucât ne dorim ca studenții să-și poată exprima opiniile la anumite probleme/propuneri pe care le au implementarea acestui serviciu este necesară oricărei aplicații deschisă publicului.

Flux: (/feedback)

5.8 Studenți

Acest serviciu se comportă ca o entitate de tip secretariat. Aici vor fi stocate și accesate datele referitoare la studenți (nume, prenume, cnp, medie, etc.)

Flux: (/studenți)

6. Aplicație desktop

Pentru a putea accesa aplicația mai ușor, vom implementa o aplicație desktop folosind framework-ul Electron. În acest mod vom putea converti frontend-ul din Angular într-o aplicație desktop.

7. Concluzie

Acest document prezintă arhitectura sistemului StudentStay, conceput pentru a asigura scalabilitate, securitate și mentenanță ușoară. Prin utilizarea

microserviciilor, sistemul este flexibil, capabil să gestioneze eficient creșterea volumului de utilizatori și să se adapteze la cerințele viitoare. Securitatea este garantată printr-un serviciu dedicat de autentificare (OAuth 2.0), protejând datele și accesul. Vom utiliza API-uri precum Amazon S3 pentru stocare sau Amazon Textract pentru extragerea textului din documente. Pentru comunicarea dintre layerele front-end și back-end, vom utiliza API-uri REST, cererile din partea clienților fiind gestionate de acestea. Astfel oferim o soluție completă și eficientă pentru bunul demers al cazării studenților în căminele TUIASI.