



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

PLANIX

Inginerie software

Autori: Tar Ivett-Dora

Toader Ana-Elena

Tăut Andreea-Bianca

Grupa: 30235

FACULTATEA DE AUTOMATICA
SI CALCULATOARE

Cuprins

1 Descriere generală a proiectului	2
1.1 Prezentarea aplicației	2
1.2 Obiectivele proiectului	2
1.3 Rezultatul final	2
2 Tehnologii utilizate	3
2.1 Backend	3
2.2 Frontend	3
3 Împărțirea pe task-uri în cadrul echipei	3
3.1 Componenta Calendar	3
3.2 Componenta Notes	4
3.3 Componenta Task	5
4 Managementul profilului de utilizator	6
4.1 Crearea unui cont de utilizator	6
4.2 Autentificarea utilizatorilor	7
5 Diagrame UML	8
5.1 Diagrama Use Case	8
5.2 Diagrama de pachete	9
5.3 Diagrama de clase	9
6 Direcții de dezvoltare viitoare	10

1 Descriere generală a proiectului

1.1 Prezentarea aplicației

Proiectul realizat constă într-o aplicație de tip **agendă digitală**, menită să ajute utilizatorii să își organizeze timpul, activitățile și informațiile personale într-un mod eficient și intuitiv. Aplicația integrează funcționalități de calendar, note, task-uri și management al profilului de utilizator, toate într-o singură platformă.

Scopul principal al aplicației este de a oferi utilizatorului un instrument complet pentru planificarea activităților zilnice, gestionarea sarcinilor și păstrarea notițelor, într-un mediu personalizat și ușor de utilizat.

1.2 Obiectivele proiectului

La începutul dezvoltării, ne-am propus realizarea unei aplicații de tip agenda digitală, care să ofere utilizatorilor posibilitatea de a-și organiza timpul și activitățile într-un mod eficient și intuitiv. Obiectivele principale ale proiectului au fost următoarele:

- dezvoltarea unei aplicații web moderne, ușor de utilizat, destinate organizării timpului prin intermediul evenimentelor, task-urilor și notițelor;
- separarea clară a părții de frontend și backend, pentru a asigura o arhitectură bine structurată, ușor de întreținut și extins;
- implementarea unui sistem de autentificare și gestionare a utilizatorilor, care să permită crearea unui cont, autentificarea în aplicație și protejarea datelor personale;
- realizarea unui sistem flexibil de adăugare, editare și ștergere a evenimentelor din calendar, a task-urilor și a notițelor, astfel încât utilizatorul să poată gestiona cu ușurință informațiile introduse;
- organizarea informațiilor prin categorii și culori, pentru a facilita identificarea rapidă a evenimentelor, task-urilor și notițelor;
- oferirea posibilității de personalizare a aplicației, prin alegerea temei vizuale și a modului de afișare;
- dezvoltarea aplicației într-un mod modular, astfel încât pe viitor să poată fi adăugate noi funcționalități, fără modificări majore ale structurii existente.

Prin aceste obiective, s-a urmărit realizarea unei aplicații complete, care să îmbine funcționalitatea cu o interfață intuitivă, adaptată nevoilor utilizatorului.

1.3 Rezultatul final

În urma implementării, aplicația dezvoltată oferă un set complet de funcționalități pentru organizarea timpului și a activităților personale. Aceasta include:

- un calendar interactiv, care permite gestionarea evenimentelor prin adăugare, editare, ștergere și filtrare;
- un modul de note, ce oferă opțiuni de creare, editare și organizare a notițelor;
- un modul de task-uri, destinat urmăririi sarcinilor zilnice și marcării acestora ca fiind finalizate;
- un sistem complet de management al profilului de utilizator, care permite crearea unui cont, autentificarea și gestionarea informațiilor personale;
- o arhitectură clară de tip backend–frontend, ușor de întreținut și extins.

În ceea ce privește personalizarea aplicației, utilizatorii pot selecta una dintre temele vizuale predefinite puse la dispoziție, fără posibilitatea de a crea teme personalizate. Această

funcționalitate oferă totuși un nivel de personalizare suficient pentru o experiență de utilizare variată.

Aplicația îndeplinește în mare parte obiectivele stabilite inițial și reprezintă o bază solidă pentru dezvoltări ulterioare și adăugarea de funcționalități suplimentare.

2 Tehnologii utilizate

2.1 Backend

Partea de backend a aplicației a fost dezvoltată folosind limbajul de programare **Java**, împreună cu framework-ul **Spring Boot**, utilizat pentru inițializarea aplicației și gestionarea componentelor acesteia.

Aplicația backend este structurată pe o arhitectură pe straturi, care asigură separarea clară a responsabilităților și o bună organizare a codului:

- **Controller** – gestionează cererile primite din partea frontend-ului;
- **Service** – conține logica de business a aplicației;
- **Repository** – se ocupă de interacțiunea cu baza de date;
- **Model** – definește entitățile aplicației;
- **DTO (Data Transfer Object)** – facilitează transferul de date între straturi;
- **Security** – gestionează autentificarea și securitatea aplicației.

Pentru persistarea datelor, aplicația utilizează o **bază de date**, prin intermediul componentelor de tip Repository, permitând stocarea și accesarea informațiilor legate de utilizatori, evenimente, task-uri și notițe.

2.2 Frontend

Partea de frontend a aplicației a fost realizată folosind:

- **JavaScript**, împreună cu biblioteca **React**;
- **JSX**, pentru definirea componentelor interfeței;
- **CSS**, pentru stilizarea aplicației.

Structura proiectului frontend este organizată modular și include:

- **components** – componente reutilizabile ale interfeței;
- **pages** – paginile principale ale aplicației;
- **assets** – resurse statice;
- **styles** – fișiere de stilizare.

Această organizare facilitează dezvoltarea și întreținerea interfeței grafice, precum și extinderea aplicației cu noi funcționalități.

3 Împărțirea pe task-uri în cadrul echipei

Proiectul a fost realizat de o echipă formată din trei membri, fiecare fiind responsabil de dezvoltarea unui modul principal al aplicației.

3.1 Componenta Calendar

Această componentă a fost realizată de Tar Ivett-Dora și permite gestionarea evenimentelor într-un calendar interactiv.

Funcționalitățile implementate sunt:

- vizualizarea calendarului pe zi;
- vizualizarea calendarului pe lună;
- vizualizarea calendarului pe an;
- adăugarea unui eveniment;
- editarea unui eveniment;
- ștergerea unui eveniment;
- atribuirea unei categorii unui eveniment;
- atribuirea unei culori pentru categorie;
- setarea duratei unui eveniment;
- filtrarea evenimentelor în funcție de categorie.

În imaginile de mai jos este prezentată interfața componentei calendar, care permite vizualizarea calendarului pe un an întreg sau pe săptămâni, crearea, editarea, ștergerea unui eveniment.

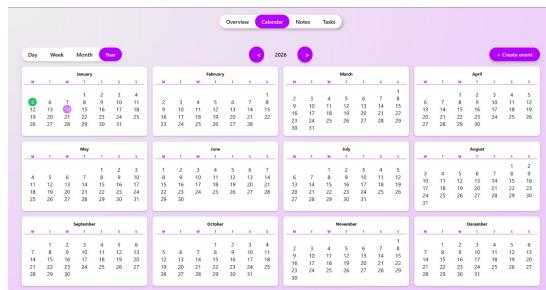


Figura 1: Vizualizare calendar pe an

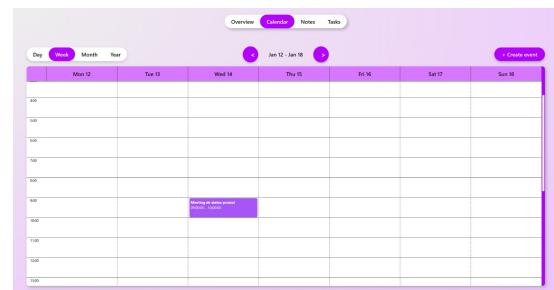


Figura 2: Calendar pe săptămâni

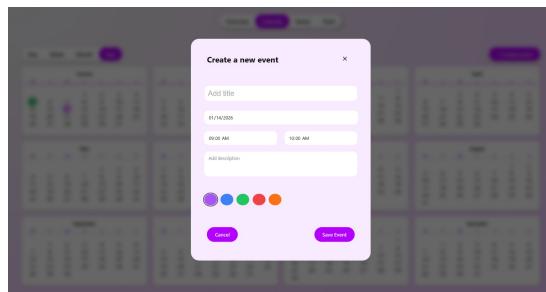


Figura 3: Creare eveniment

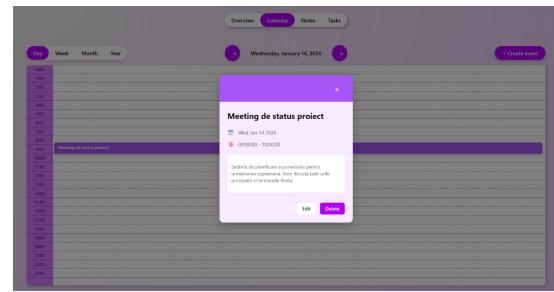


Figura 4: Editare/ ștergere eveniment

3.2 Componenta Notes

Această componentă a fost realizată de Toader Ana-Elena și este destinat gestionării notișelor personale.

Funcționalitățile disponibile includ:

- adăugarea unei note;
- editarea unei note;
- ștergerea unei note;
- atribuirea unei categorii unei note;
- atribuirea unei culori pentru note;
- utilizarea unor fonturi de bază pentru stilizarea textului.

În imaginile de mai jos este prezentată interfața componentei de notes, care permite vizualizarea, crearea, editarea, ștegerea unei notițe, selectarea unei categorii și adăugarea la favorite.

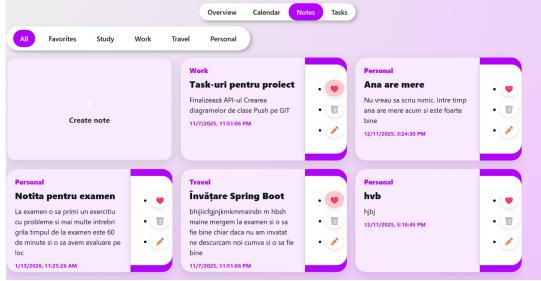


Figura 5: Vizualizare notițe

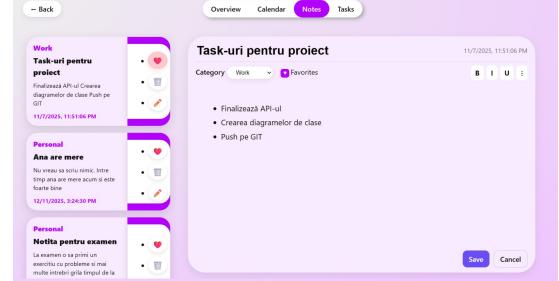


Figura 6: Creare/editare notiță

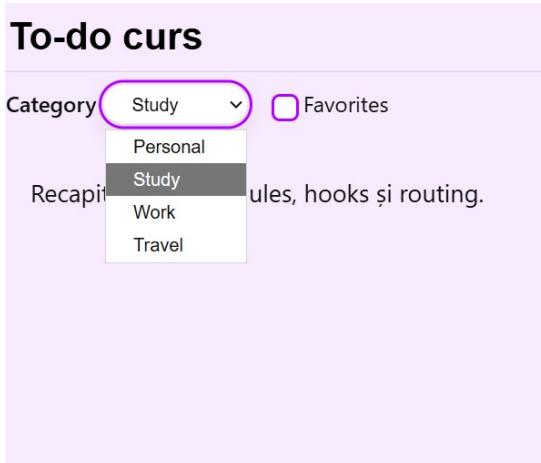


Figura 7: Selectare categorie

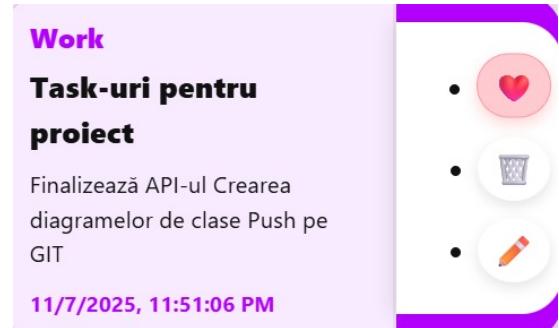


Figura 8: Favorite/Ștergere

3.3 Componenta Task

Această componentă a fost realizată de Tăut Andreea- Bianca și se ocupă de gestionarea sarcinilor zilnice.

Funcționalitățile implementate sunt:

- adăugarea unui task;
- editarea unui task;
- ștergerea unui task;
- atribuirea unei categorii unui task;
- atribuirea unei culori unui task;
- marcarea unui task ca fiind finalizat;
- filtrarea task-urilor în funcție de ziua selectată din calendar;
- filtrarea task-urilor în funcție de luna selectată.

În imaginile de mai jos este prezentată interfața componentei de task-uri, care permite vizualizarea, crearea, editarea și gestionarea sarcinilor utilizatorului.

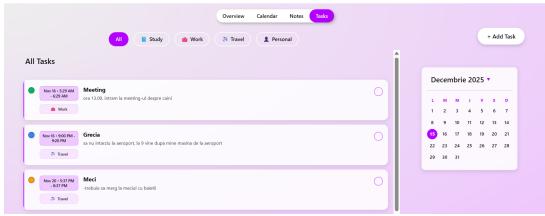


Figura 9: Task-uri



Figura 10: Task marcat ca finalizat

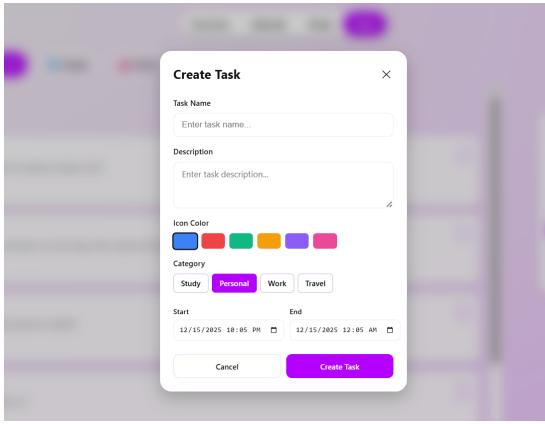


Figura 11: Creare task

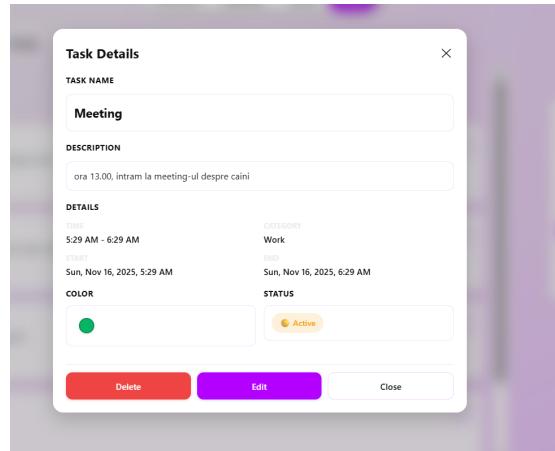


Figura 12: Editare/Detalii task

4 Managementul profilului de utilizator

Aplicația **PLANIX** oferă un sistem de autentificare securizat, care permite utilizatorilor să își creeze un cont personal și să se conecteze ulterior pentru a accesa funcționalitățile aplicației. Acest mecanism asigură protejarea datelor și personalizarea experienței fiecărui utilizator.

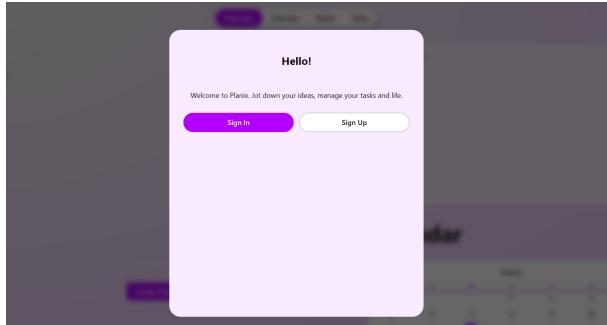


Figura 13: Pagina de pornire

4.1 Crearea unui cont de utilizator

Pentru a putea utiliza aplicația, utilizatorul trebuie să își creeze un cont prin intermediul formularului de înregistrare. Procesul de creare a contului presupune introducerea următoarelor informații:

- adresa de e-mail;
- numele și prenumele;

- un nume de utilizator (nickname);
- o parolă.

Parola este validată conform unor reguli de securitate, pentru a asigura un nivel minim de protecție a contului. Aceasta trebuie să îndeplinească următoarele condiții:

- să aibă o lungime minimă de 8 caractere;
- să conțină cel puțin o literă mare;
- să includă cel puțin o cifră sau un caracter special.

Înainte de a fi stocată în baza de date, parola este criptată utilizând algoritmul **BCrypt**, prevenind astfel accesul neautorizat în cazul compromiterii bazei de date.

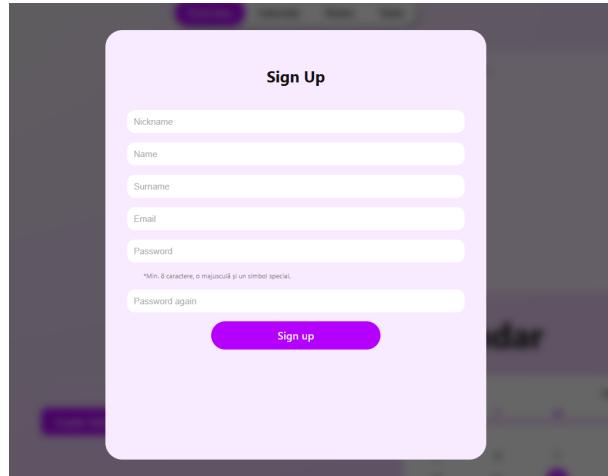


Figura 14: Crearea unui cont de utilizator

4.2 Autentificarea utilizatorilor

După crearea contului, utilizatorul se poate autentifica în aplicație folosind adresa de e-mail și parola asociată contului. În urma autentificării cu succes, serverul generează un **token de tip JWT (JSON Web Token)**, care este transmis către client și utilizat pentru autorizarea cererilor ulterioare.

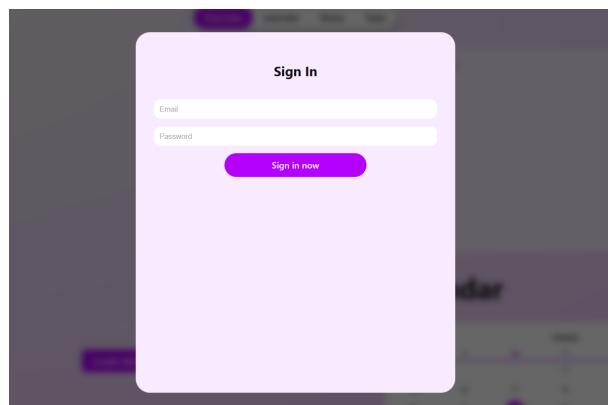


Figura 15: Logarea utilizatorului

5 Diagrame UML

Pentru a evidenția structura și funcționalitățile aplicației, au fost realizate mai multe diagrame UML. Acestea oferă o perspectivă clară asupra modului de utilizare a aplicației, a organizării componentelor și a relațiilor dintre clasele principale.

Diagramele au fost realizate utilizând platforma **Draw.io** și au rolul de a sprijini înțelegerea arhitecturii și a funcționării generale a aplicației.

5.1 Diagrama Use Case

Diagrama Use Case ilustrează interacțiunea dintre utilizator și aplicație, evidențiind principalele funcționalități disponibile. Aceasta descrie acțiunile pe care utilizatorul le poate realiza, precum autentificarea, gestionarea evenimentelor din calendar, administrarea task-urilor și a notițelor.

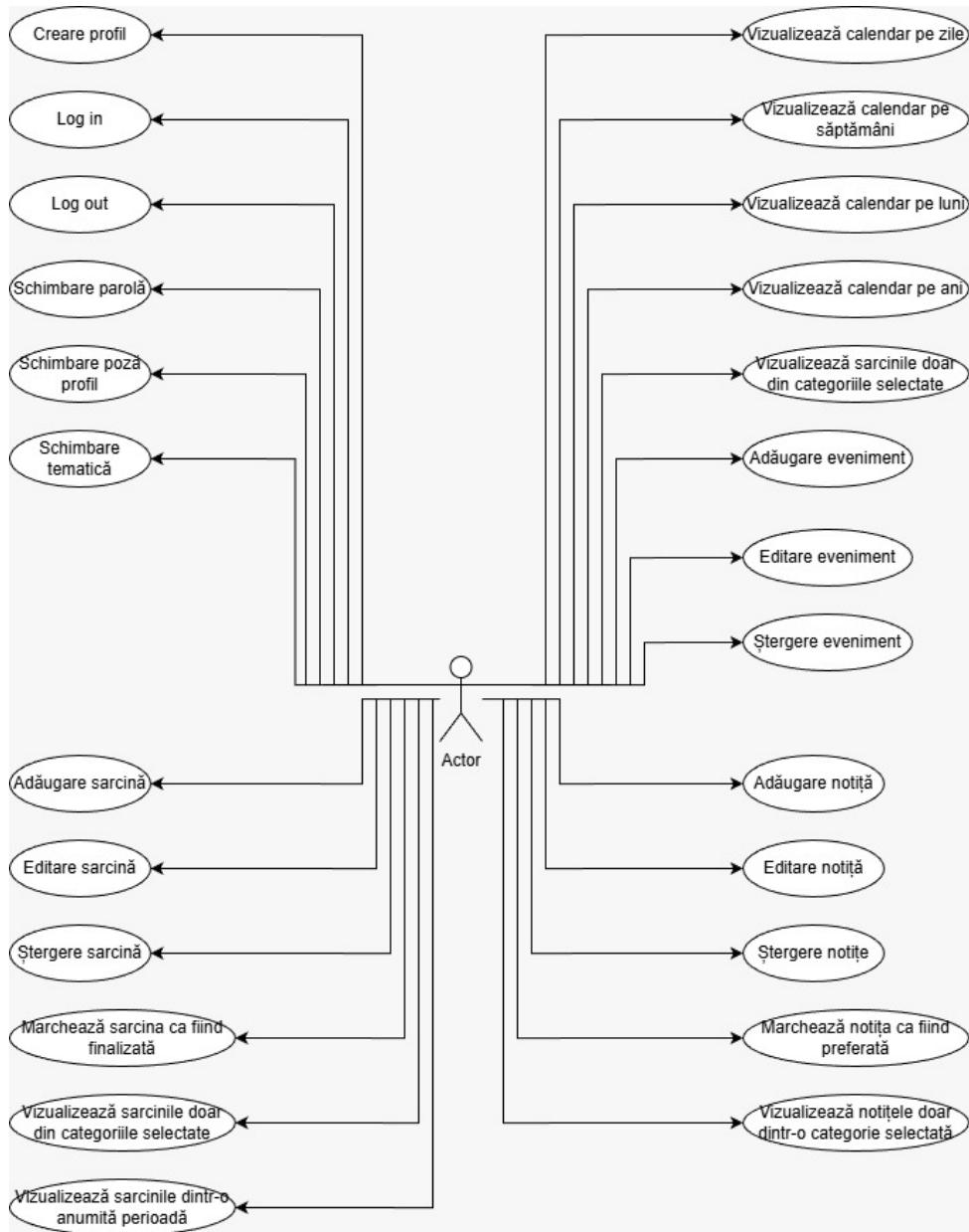


Figura 16: Diagrama Use Case a aplicației PLANIX

5.2 Diagrama de pachete

Diagrama de pachete prezintă organizarea aplicației pe module și evidențiază separarea clară dintre componentele de frontend și backend. Aceasta reflectă arhitectura modulară a aplicației și relațiile dintre diferitele pachete, facilitând înțelegerea structurii generale a proiectului.

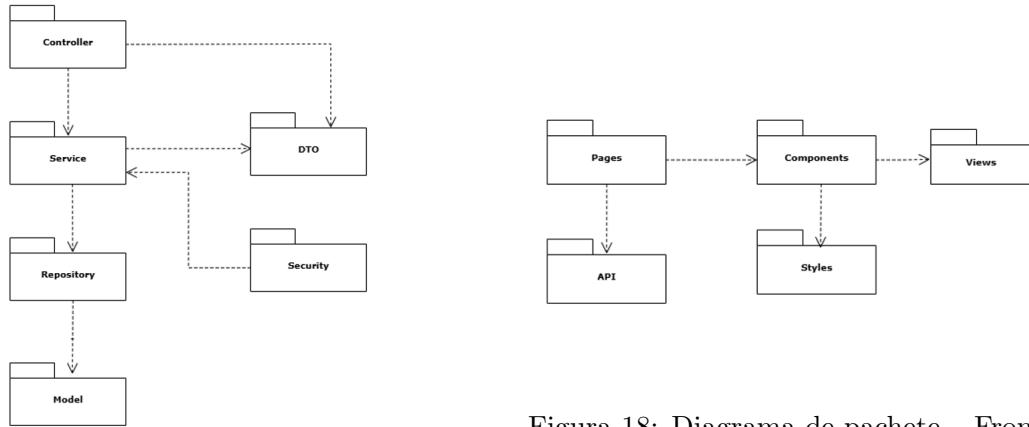


Figura 18: Diagrama de pachete – Frontend

Figura 17: Diagrama de pachete – Backend

5.3 Diagrama de clase

Diagrama de clase descrie structura statică a aplicației, prezentând clasele principale, attributele acestora și relațiile dintre ele. Aceasta oferă o imagine de ansamblu asupra modului în care sunt modelate entitățile aplicației și asupra interacțiunilor dintre componentebackend.

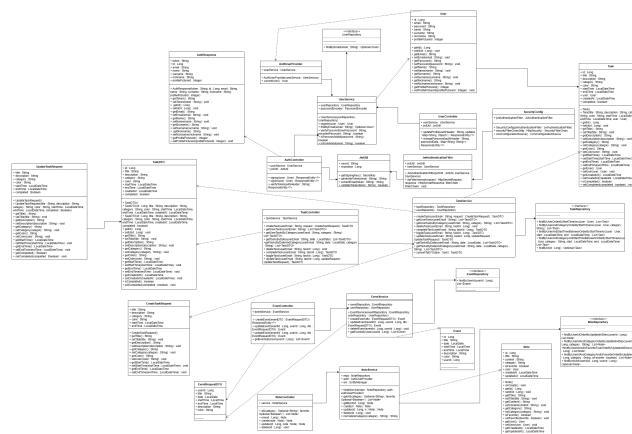


Figura 19: Diagrama de clase a aplicației PLANIX

6 Direcții de dezvoltare viitoare

Aplicația dezvoltată oferă o bază solidă pentru extinderi ulterioare, putând fi îmbunătățită prin adăugarea unor funcționalități suplimentare menite să crească nivelul de interactivitate și personalizare. Printre direcțiile de dezvoltare viitoare se numără:

- implementarea unui sistem de notificări pentru evenimente și task-uri, care să informeze utilizatorul în timp real sau la intervale prestabilite;
- salvarea și sincronizarea datelor în cloud, astfel încât informațiile să fie accesibile de pe mai multe dispozitive;
- adăugarea suportului pentru mai multe limbi, pentru a crește accesibilitatea aplicației;
- generarea de statistici și rapoarte privind productivitatea utilizatorului, precum timpul alocat task-urilor sau numărul de activități finalizate;
- dezvoltarea unei versiuni mobile a aplicației, adaptată pentru dispozitivele mobile;
- extinderea opțiunilor de personalizare a aplicației, prin posibilitatea ca utilizatorii să își creeze propriile teme vizuale, să își aleagă culorile și stilurile preferate, precum și să personalizeze modul de afișare al componentelor.