



Tema 3 - Sablonul arhitectural MVC

Proiectare Software

Student: Mirisan Octavian

Grupa: 30231

FACULTATEA DE AUTOMATICA
SI CALCULATOARE

Mai 2025

Cuprins

1	Enuntul Problemei	2
2	Instrumente utilizate	2
3	Justificarea limbajului de programare	2
4	Descriere diagrame UML	3
4.1	Diagrama de Use Case	3
4.2	Diagramele de activitate	4
4.3	Diagrama de pachete	11
4.4	Diagrama de clase	12
4.5	Diagrama entitate-relatie	13
4.6	Diagramele de secventa	14
5	Descriere aplicatie	22
5.1	Interfata artistilor	22
5.2	Interfata operelor de arta	23
5.3	Interfata statisticilor	24

1 Enuntul Problemei

Problema 5

Dezvoltați o aplicație soft care poate fi utilizată într-o galerie de artă pentru gestiunea operelor de artă expuse. Softul va permite:

- ❖ Adăugarea, ștergerea și actualizarea artiștilor care au expuse opere de artă în galeria de artă;
- ❖ Vizualizarea listei tuturor artiștilor care au expuse opere de artă în galeria de artă (pentru fiecare artist se va afișa numele, data nașterii, locul nașterii, naționalitatea, o fotografie și lista tuturor operelor de artă realizate de artist și expuse în acest muzeu);
- ❖ Căutarea unui artist după nume;
- ❖ Adăugarea, ștergerea și actualizarea operelor de artă expuse în galeria de artă;
- ❖ Vizualizarea listei tuturor operelor de artă expuse în galeria de artă sortate după artist (vizualizarea include și redarea unor imagini cu operele de artă; între 1 și 3 imagini pentru fiecare operă de artă);
- ❖ Căutarea unei opere de artă după titlu;
- ❖ Filtrarea listei operelor de artă după următoarele criterii: artist, preț sau tipul operei de artă;
- ❖ Salvarea listei cu operele de artă expuse în fișiere de tip csv și doc;
- ❖ Vizualizarea unor statistici legate de operele de artă utilizând grafice.

Interfața grafică a aplicației va fi disponibilă în cel puțin 3 limbi de circulație internațională.

Figura 1: Enuntul problemei

2 Instrumente utilizate

Instrumentele utilizate în procesul de dezvoltare a aplicatiei software includ:

- **JavaFX:** framework-ul JavaFX a fost utilizat pentru crearea interfeței grafice a aplicatiei, fiind ideal pentru aplicații desktop interactive
- **Scene Builder:** a fost folosit pentru proiectarea interfeței grafice în format FXML. Acest instrument permite crearea rapidă și eficientă a layout-urilor pentru interfețele dedicate artiștilor și operelor de artă.
- **IntelliJ IDEA:** un IDE specializat pe limbajul Java, utilizat pentru scrierea, depanarea și testarea codului aplicatiei. Funcționalitățile sale avansate au îmbunatatit productivitatea în timpul dezvoltării.
- **SQLite:** a fost ales ca sistem de gestiune a bazelor de date relationale datorită simplitatii și portabilității sale. Baza de date a fost utilizată pentru persistența artiștilor, operelor de artă, precum și imaginilor asociate operelor.

3 Justificarea limbajului de programare

Limbajul de programare ales pentru implementarea aplicatiei este **Java**, iar pentru interfața grafică s-a utilizat **JavaFX**.

Motivele pentru care am ales aceste limbi sunt:

- **Suport pentru programare OOP:** structura aplicatiei, bazată pe pachete precum **Model, Repository, View, Controller**, se aliniaza în mod natural cu paradigma object oriented și cu tot ceea ce presupune ea (mostenire, implementare, etc.)
- **Integrare cu JavaFX:** JavaFX, ca parte a ecosistemului Java, a fost alegerea naturală pentru crearea interfeței grafice. Aceasta oferă suport pentru componente vizuale complexe, cum ar fi tabelele (pentru afișarea artiștilor și a operelor), ImageView-urile (pentru afișarea imaginilor în fereastra), TextField-uri, grafice, etc.
- **Gestionarea bazelor de date:** Java oferă suport nativ pentru lucrul cu baze de date, prin **JDBC** (Java Database Connectivity). În acest proiect, JDBC a fost folosit pentru a interacționa cu SQLite, permitând astfel operații CRUD.

4 Descriere diagrame UML

4.1 Diagrama de Use Case

Diagrama de use case descrie interactiunile utilizatorului cu aplicatia de gestionare a unei galerii de arta. Utilizatorul, reprezentat ca actor, poate efectua actiuni precum adaugarea, stergerea si actualizarea artistilor si operelor de arta, precum si vizualizarea listelor acestora. De asemenea, utilizatorul are posibilitatea de a cauta opere de arta, de a filtra liste si de a salva liste de opere pentru export. Sistemul, denumit "Aplicatie", incapsuleaza toate aceste functionalitati, oferind o interfata centralizata pentru gestionarea datelor galeriei.

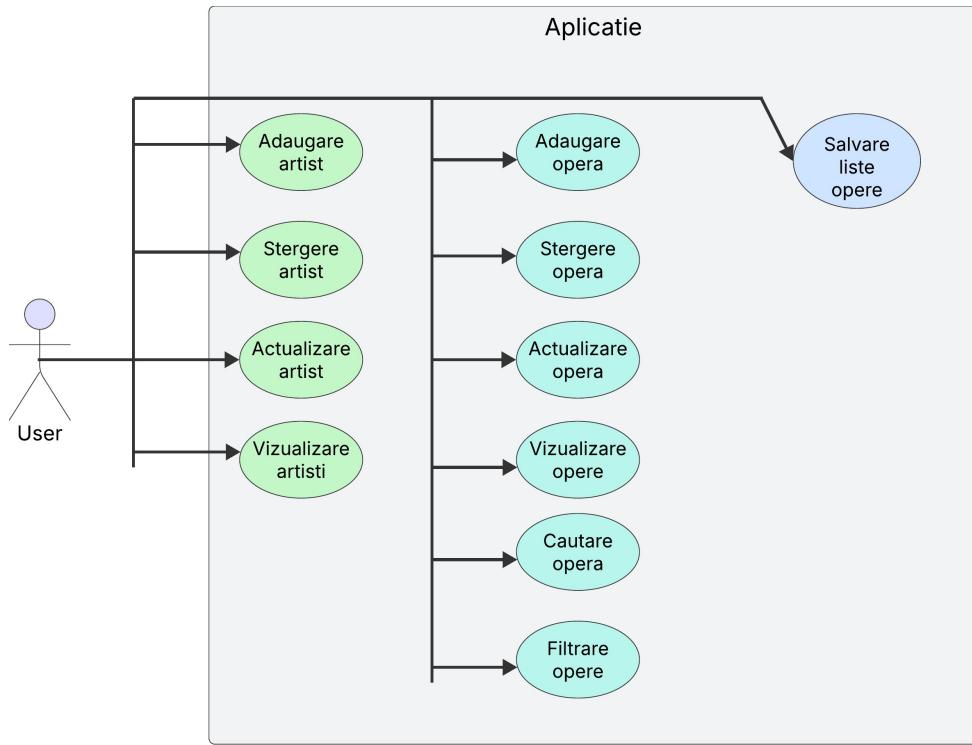


Figura 2: Diagrama de Use Case

4.2 Diagramele de activitate

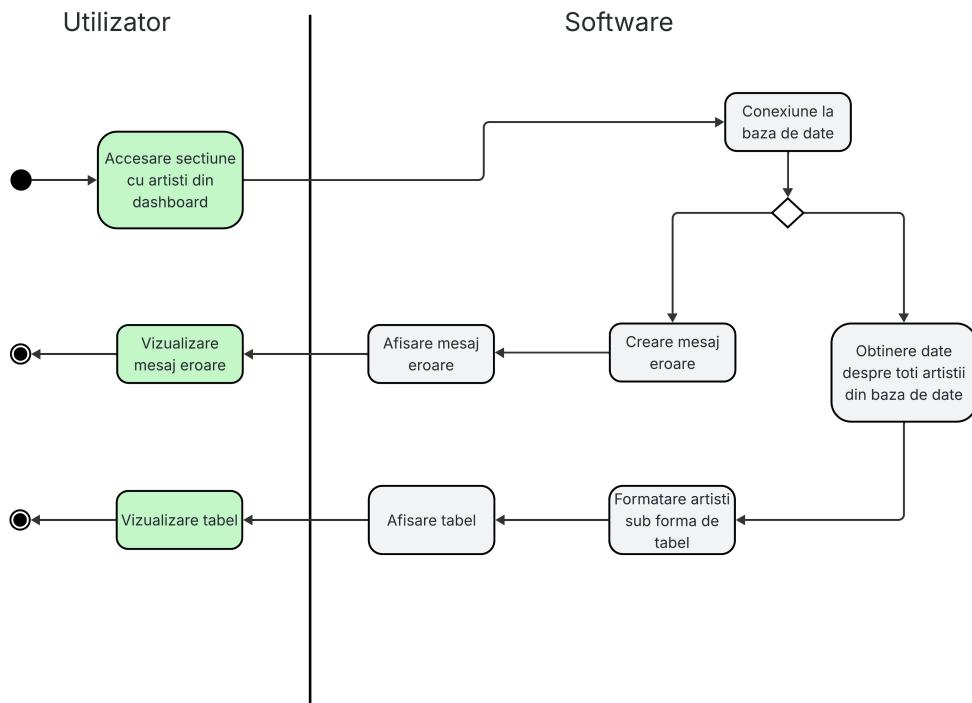


Figura 3: Vizualizare artiști

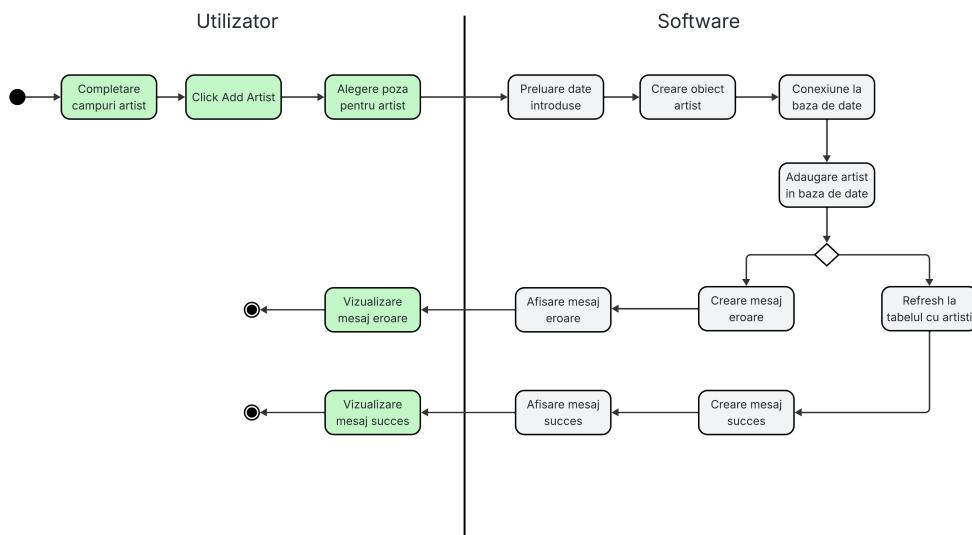


Figura 4: Adaugare artișt în baza de date.

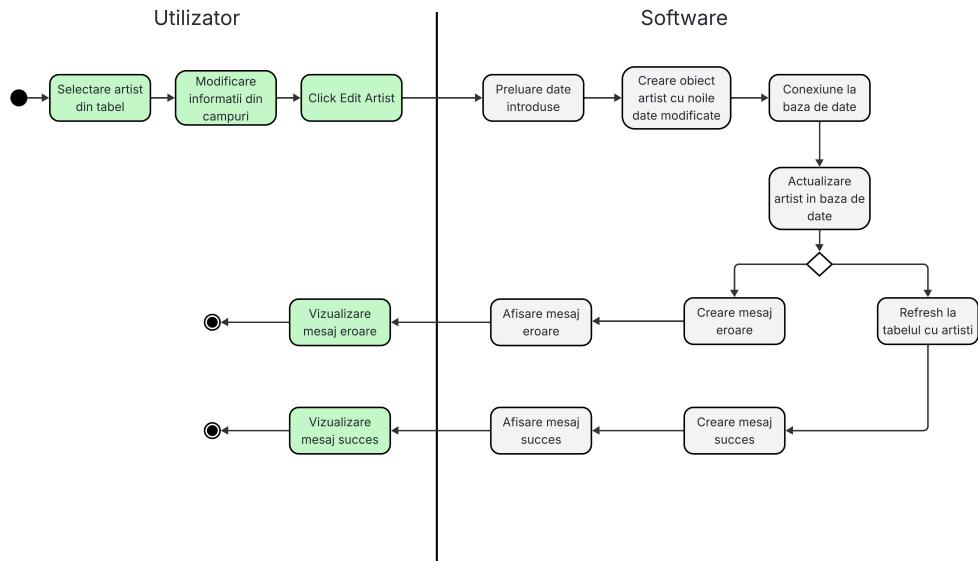


Figura 5: Editare artist in baza de date.

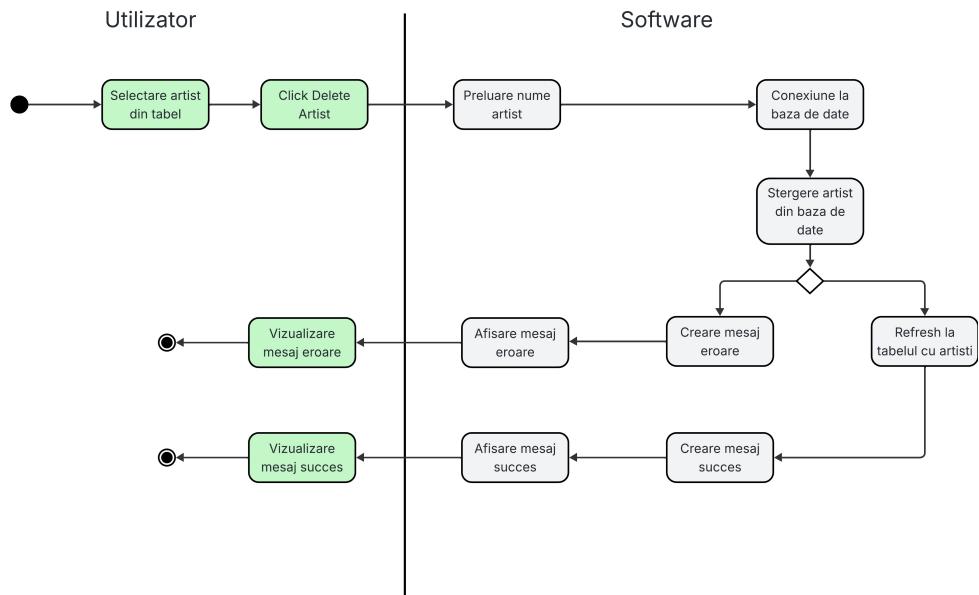


Figura 6: Stergere artist din baza de date.

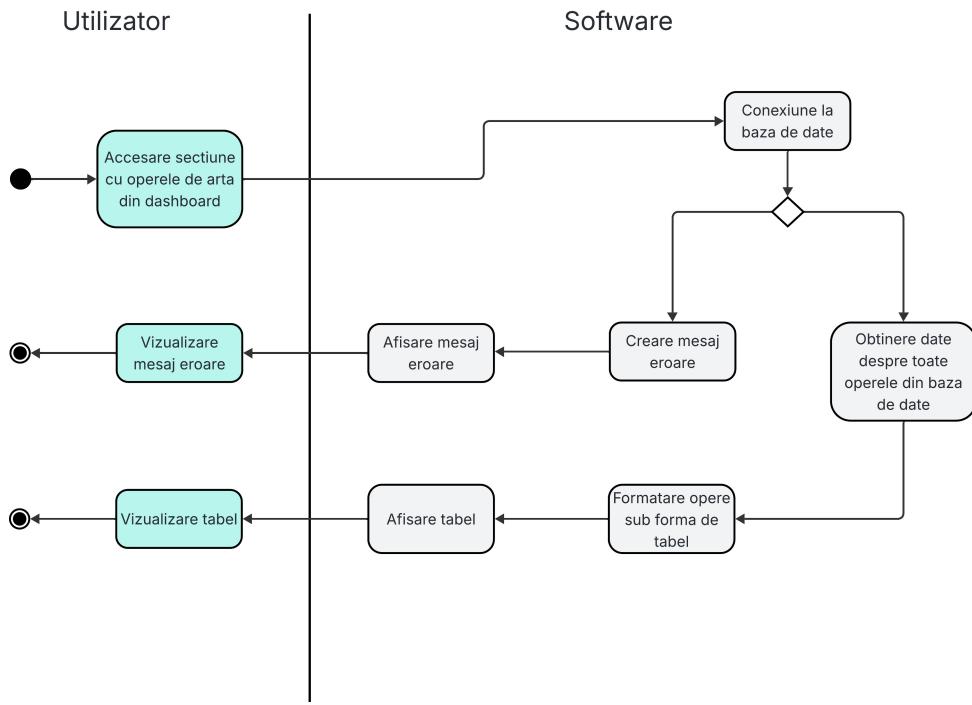


Figura 7: Vizualizare opere de arta

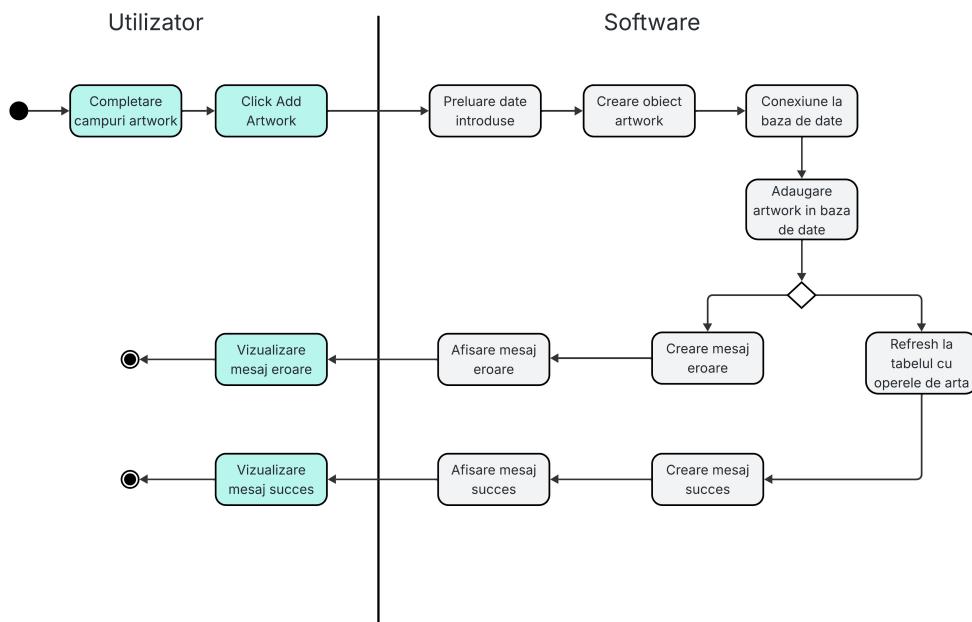


Figura 8: Adaugare opera de arta in baza de date.

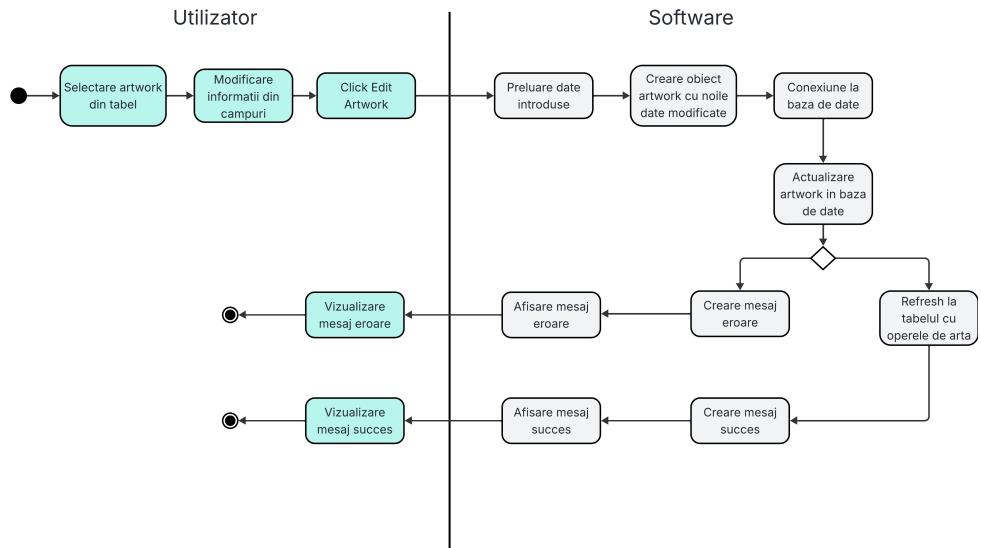


Figura 9: Editare opera de arta in baza de date.

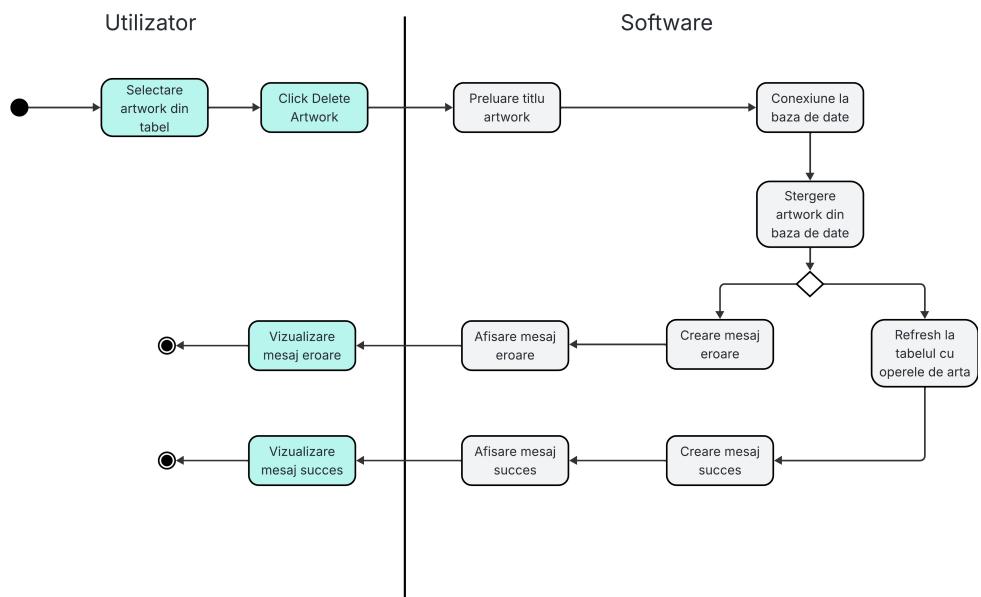


Figura 10: Stergere opera de arta din baza de date.

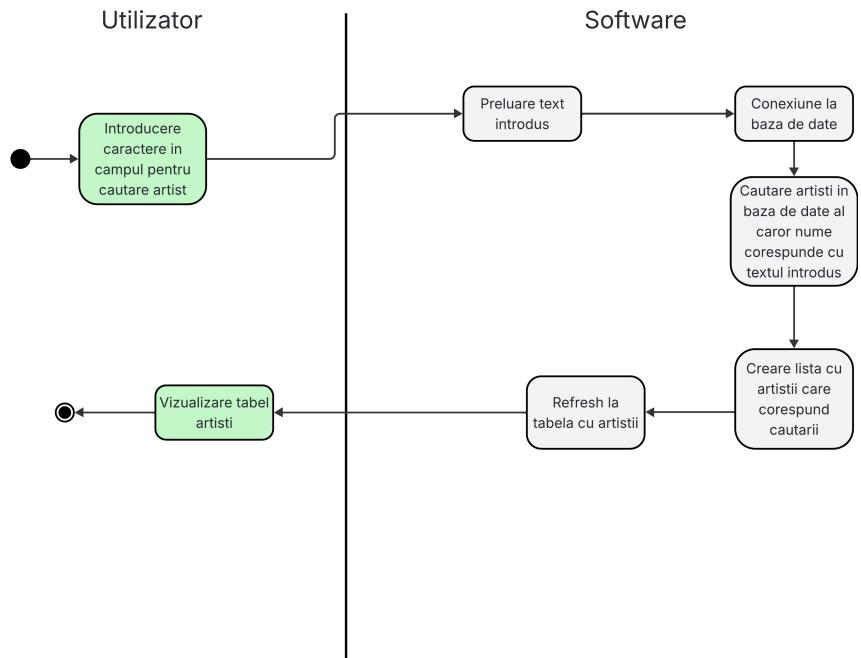


Figura 11: Cautare artist după nume.

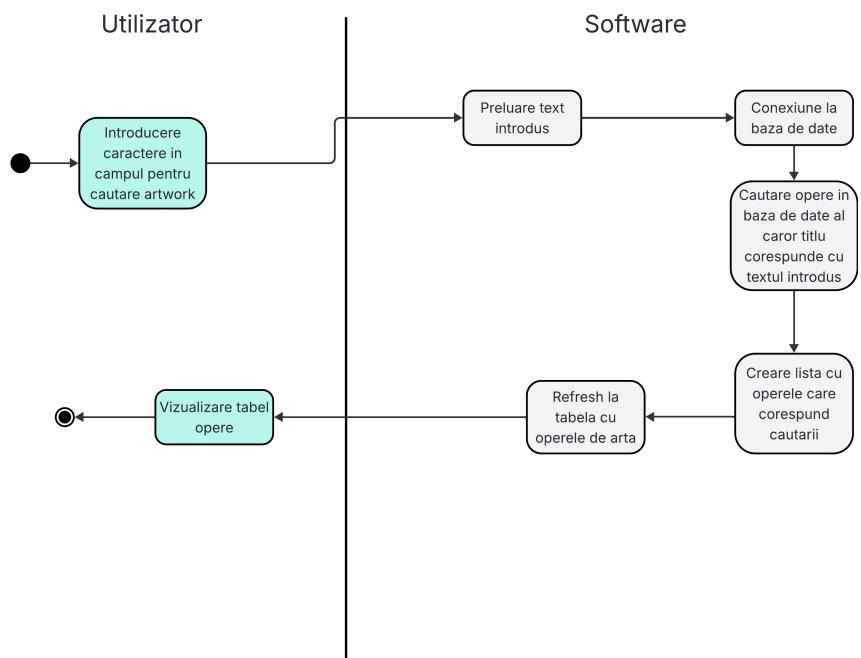


Figura 12: Cautare opera de artă după titlu.

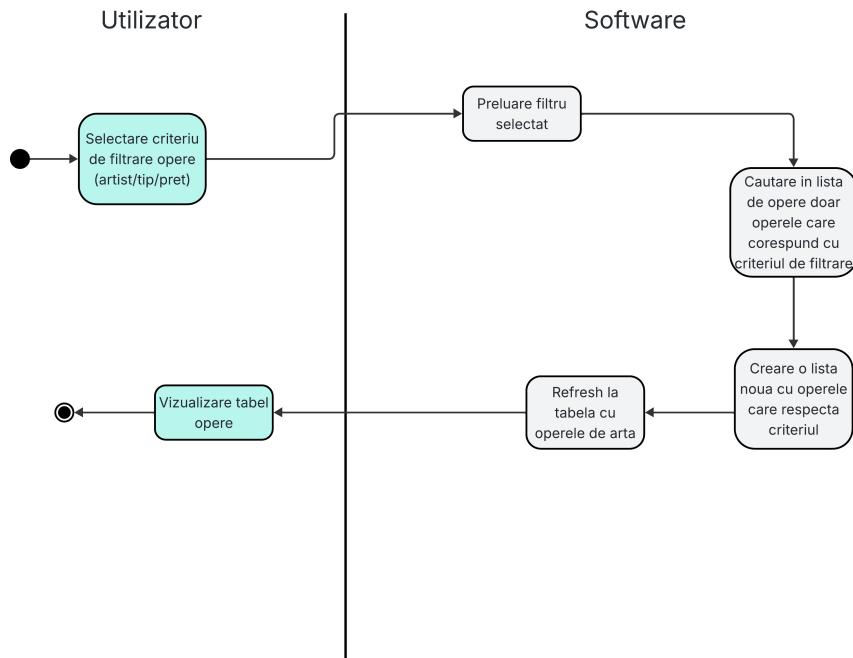


Figura 13: Filtrare opere de artă

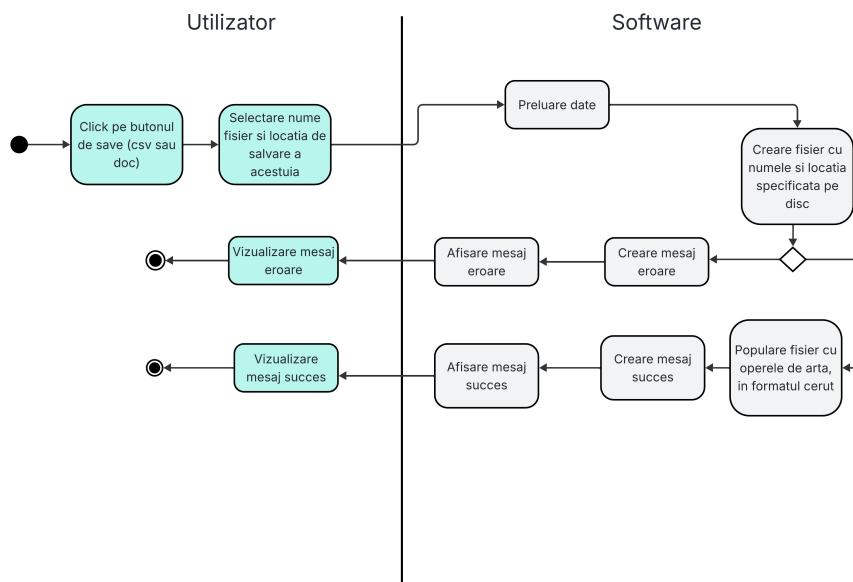


Figura 14: Export opere de artă în format .csv sau .doc

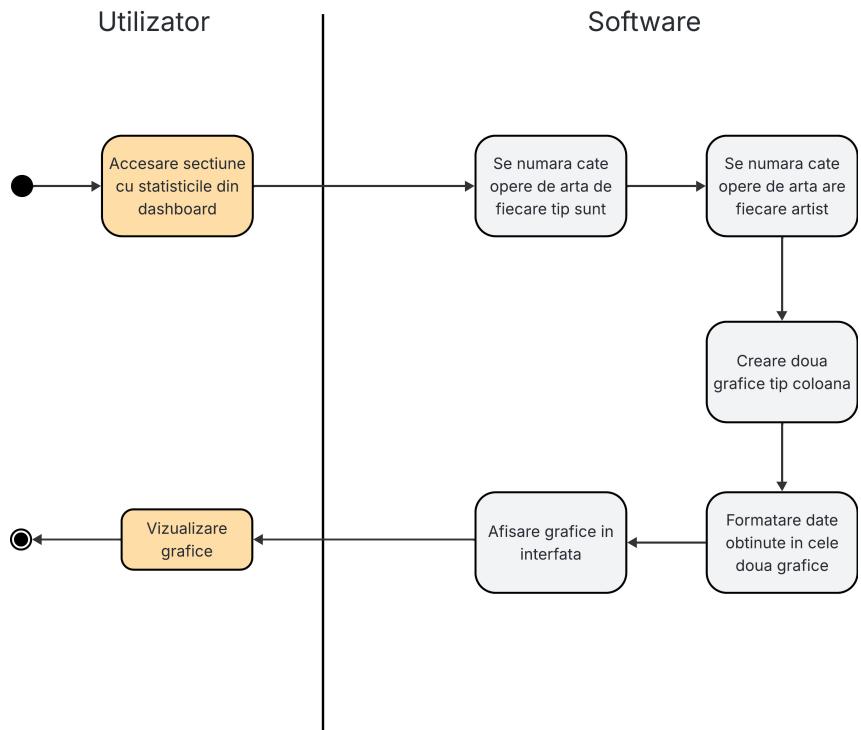


Figura 15: Vizualizare statistici

4.3 Diagrama de pachete

Diagrama de pachete ilustreaza organizarea structurala a aplicatiei de gestionare a galeriei de arta, evidentiind dependentele dintre pachete.

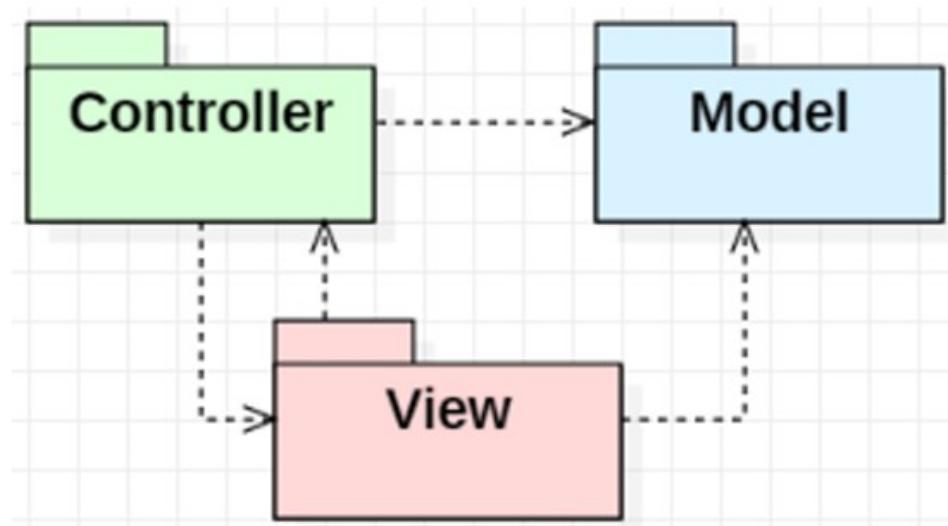


Figura 16: Diagrama de pachete a aplicatiei

4.4 Diagrama de clase

Explicatii:

- **Pachetul Model:** este responsabil de definirea entităților fundamentale ale aplicației: **Artist**, **Artwork** și **ArtworkImage**. Aceste clase descriu structura datelor care sunt manipulate în aplicație.
- **Pachetul Repository:** conține clasele care interacționează direct cu baza de date. Fiecare entitate are asociat un repository specific (**ArtistRepo**, **ArtworkRepo**, **ArtworkImageRepo**) care oferă operații CRUD și metode de interogare.
- **Pachetul ViewModel:** funcționează ca un mediator între Model și View. Clasa **GalleryViewModel** gestionează logica de transformare a datelor, apeleză repository-urile și notifică View-ul folosind un mecanism propriu de tip **Observable**.
- **Pachetul View:** gestionează afișarea și interacțiunea cu utilizatorul. Clasa **GalleryView** definește toate componentele UI declarate în fișierul FXML și implementează interfața **Observer**, actualizând automat interfața grafică atunci când modelul de date se modifică.
- **Pachetul Controller:** conține clasa **GalleryController**, care preia acțiunile utilizatorului din View și apeleză metodele corespunzătoare din ViewModel. Acest pachet conține logica de legătură între UI și date, fără să conțină cod specific de afișare.
- **Pachetul Observer:** implementează manual şablonul de proiectare **Observer**, folosind interfața **Observer** și clasa abstractă **Observable**, oferind astfel un mecanism de notificare între ViewModel și View.

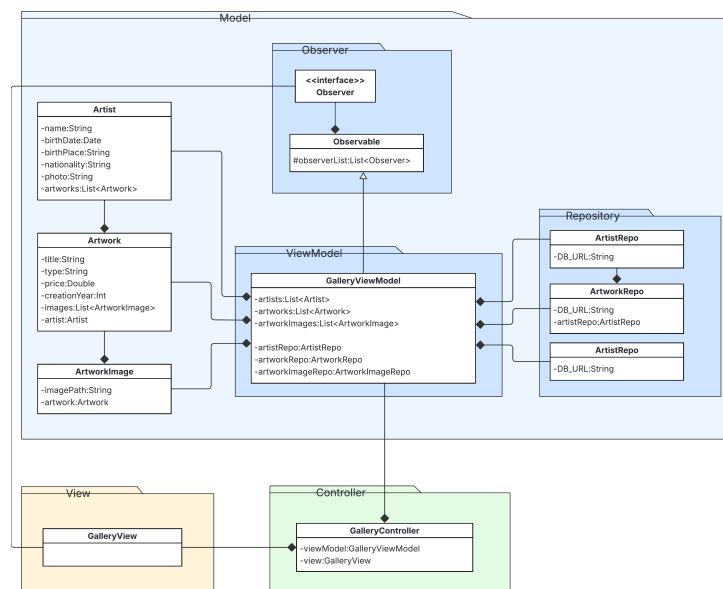


Figura 17: Diagrama de clase

4.5 Diagrama entitate-relatie

Relatiile intre entitati sunt in felul urmator:

- Un artist poate crea **una sau mai multe** opere de arta.
- O opera de arta apartine **cel mult** unui artist.
- O opera de arta contine **una sau mai multe** imagini.
- O imagine este asociata **unei singure** opere de arta.

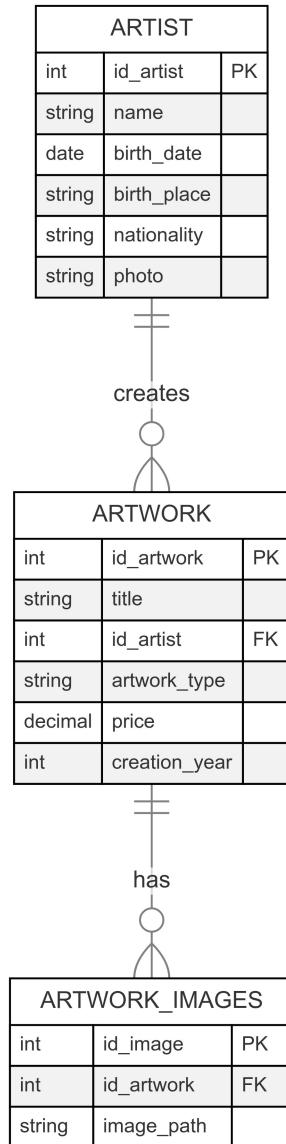


Figura 18: Diagrama entitate-relatie

4.6 Diagramele de sevanta

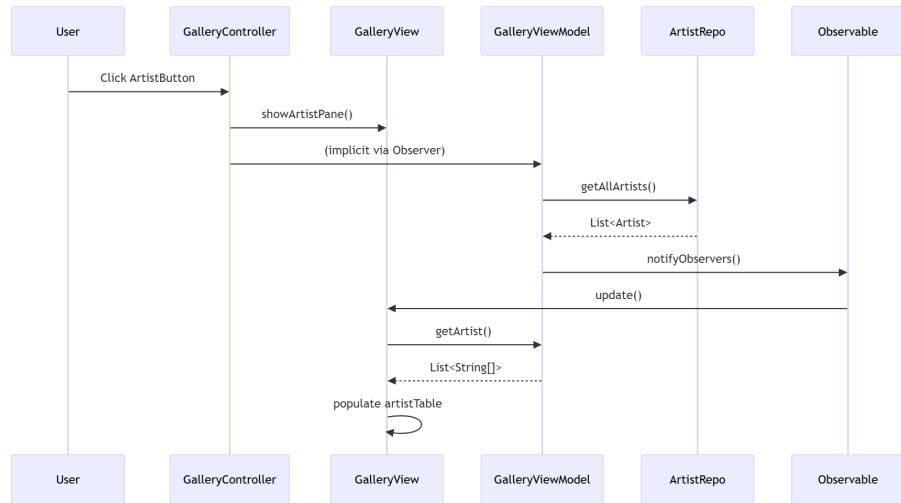


Figura 19: Vizualizare artiști

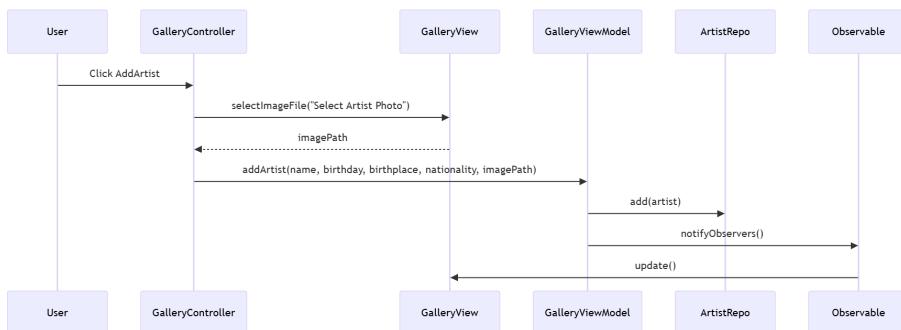


Figura 20: Adaugare artișt în baza de date.

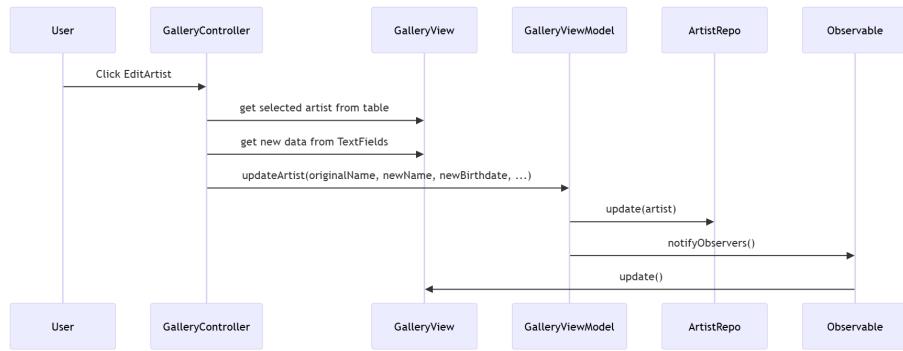


Figura 21: Editare artist in baza de date.

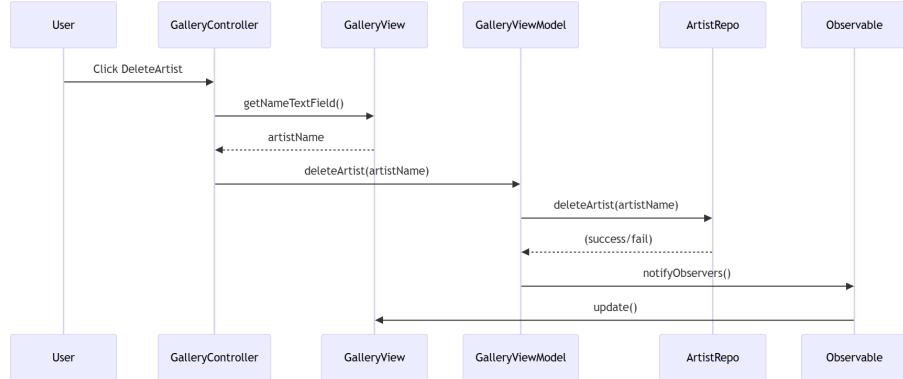


Figura 22: Stergere artist din baza de date.

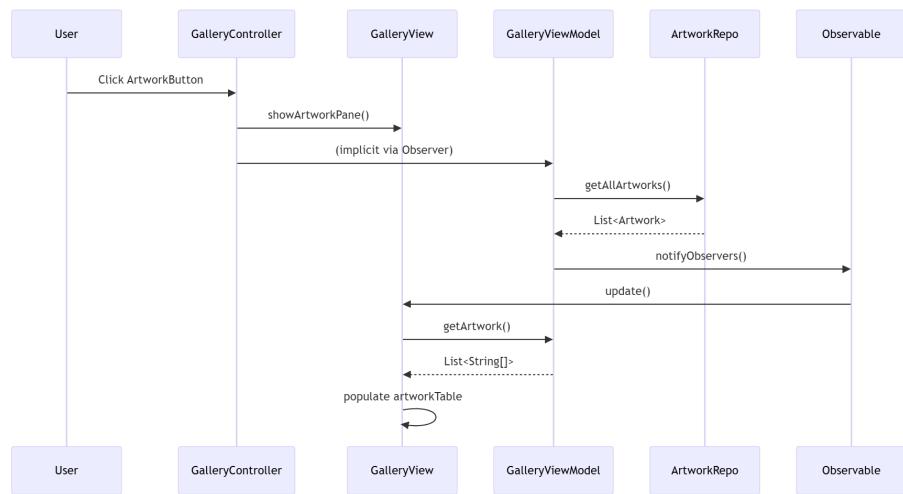


Figura 23: Vizualizare opere de artă

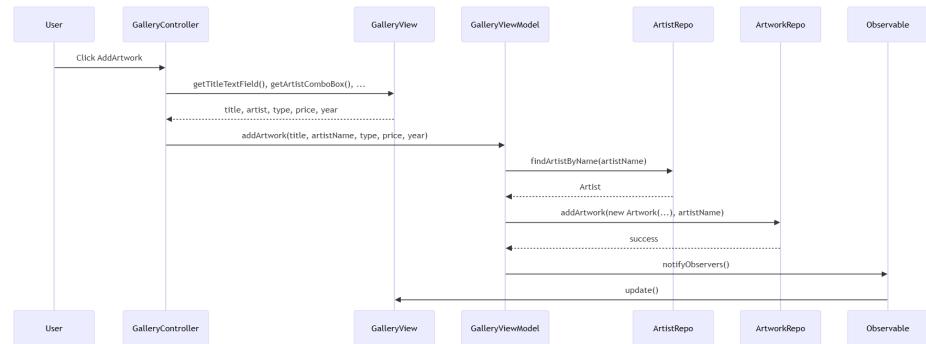


Figura 24: Adaugare opera de artă în bază de date.

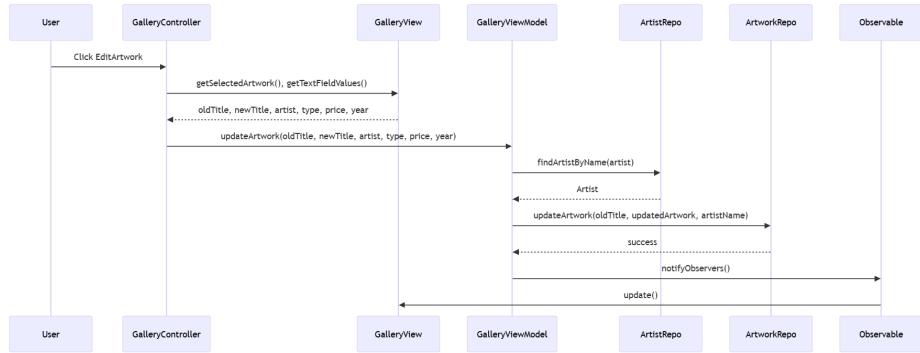


Figura 25: Editare opera de arta in baza de date.

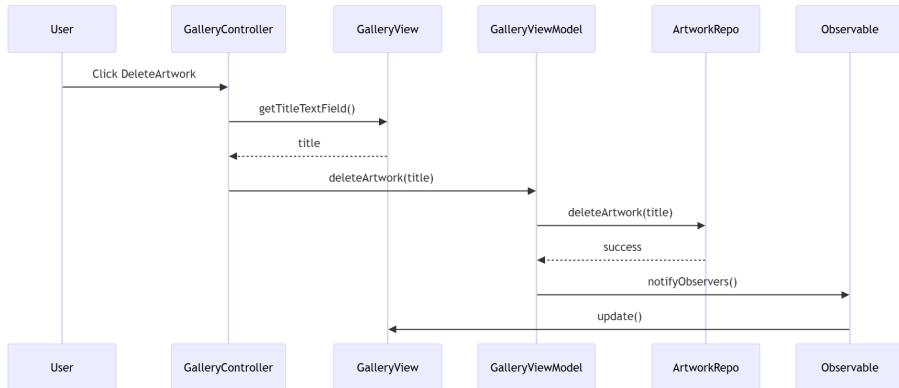


Figura 26: Stergere opera de arta din baza de date.

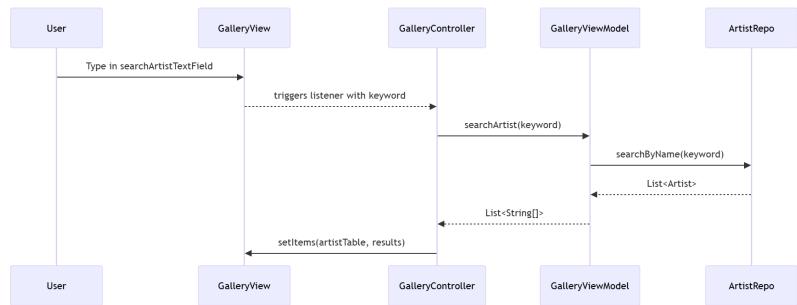


Figura 27: Cautare artist dupa nume.

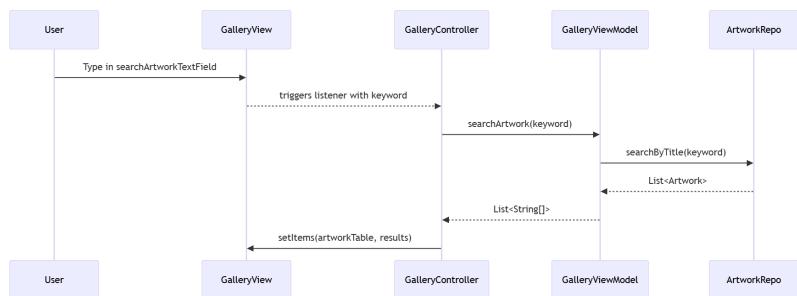


Figura 28: Cautare opera de artă după titlu.

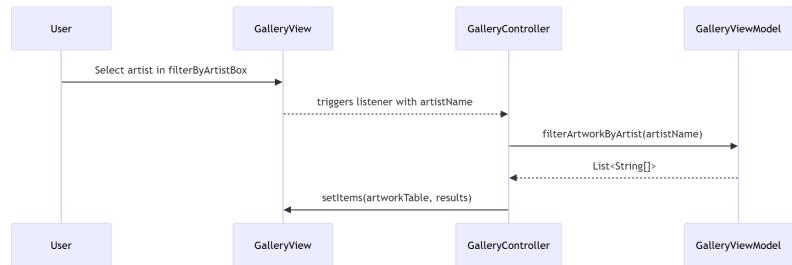


Figura 29: Filtrare opere de arta după artist

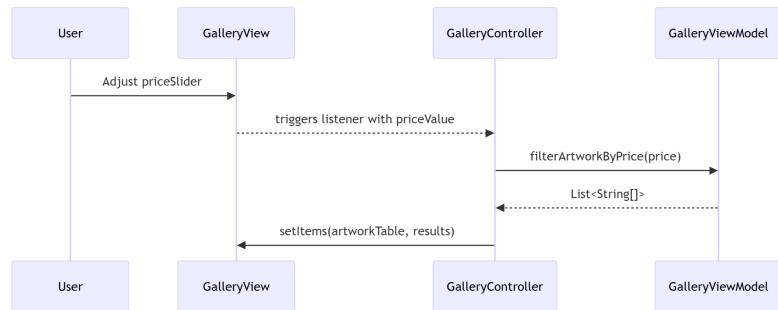


Figura 30: Filtrare opere de arta după pret

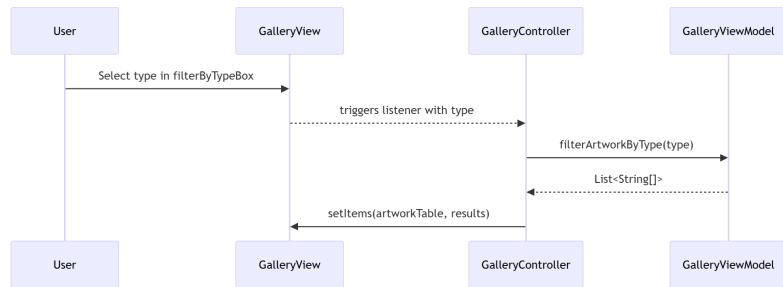


Figura 31: Filtrare opere de arta după tip

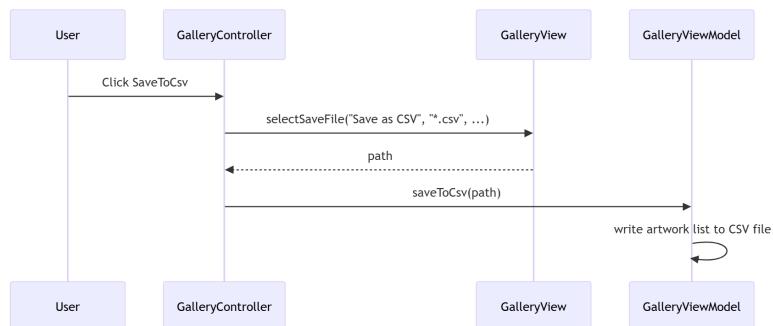


Figura 32: Export opere de artă în format .csv

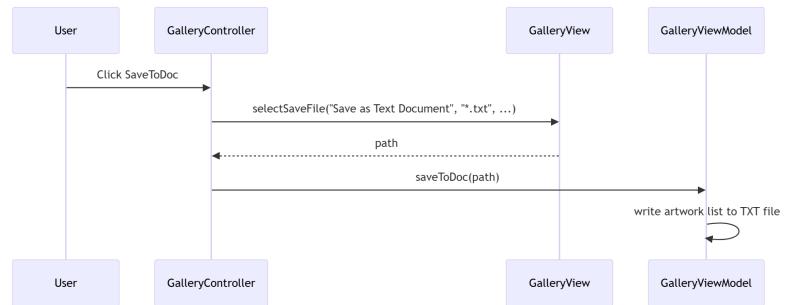


Figura 33: Export opere de arta in format .doc

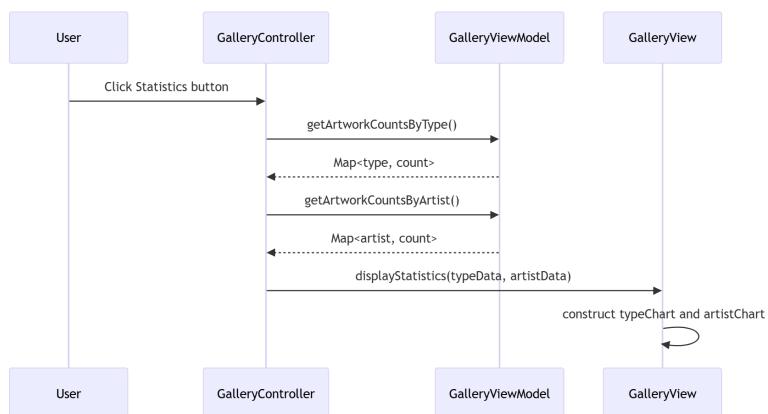


Figura 34: Vizualizare statistici

5 Descriere aplicatie

5.1 Interfata artistilor

Acest dashboard reprezintă o interfață grafică simplă și eficientă concepută pentru gestionarea informațiilor despre artiști. În partea stângă, utilizatorii au acces la un meniu cu opțiunile „Artiști” și „Opere de artă”, iar secțiunea „Artiști” este activă. Partea superioară include un formular intuitiv unde se pot introduce detalii esențiale despre un artist, cum ar fi numele (ex. „Vincent van Gogh”), data nașterii (ex. „3/2/1853”), locul nașterii (ex. „Olanda”) și naționalitatea (ex. „olandeză”). Formularul este completat de trei butoane practice: „ADD” pentru adăugarea unui nou artist, „EDIT” pentru modificări și „DELETE” pentru ștergerea unui artist, oferind astfel control total asupra datelor. În dreapta, este afișată o imagine reprezentativă a artistului selectat, ceea ce face experiența vizuală mai atractivă.

Sub formular, un tabel bine structurat listează artiștii înregistrați, incluzând coloane precum numele, data și locul nașterii, naționalitatea și o listă a operelor lor. De exemplu, Vincent van Gogh apare cu data „1853-03-02”, „Olanda” ca loc de naștere și opere precum „Starry Night” sau „Sunflower Field”. Tabelul mai include și alți artiști renumiți, precum Pablo Picasso, Claude Monet, Leonardo da Vinci, Auguste Rodin și Henri Cartier-Bresson, fiecare cu informațiile și operele lor asociate. Această organizare facilitează căutarea rapidă și gestionarea eficientă a bazei de date a artistilor.

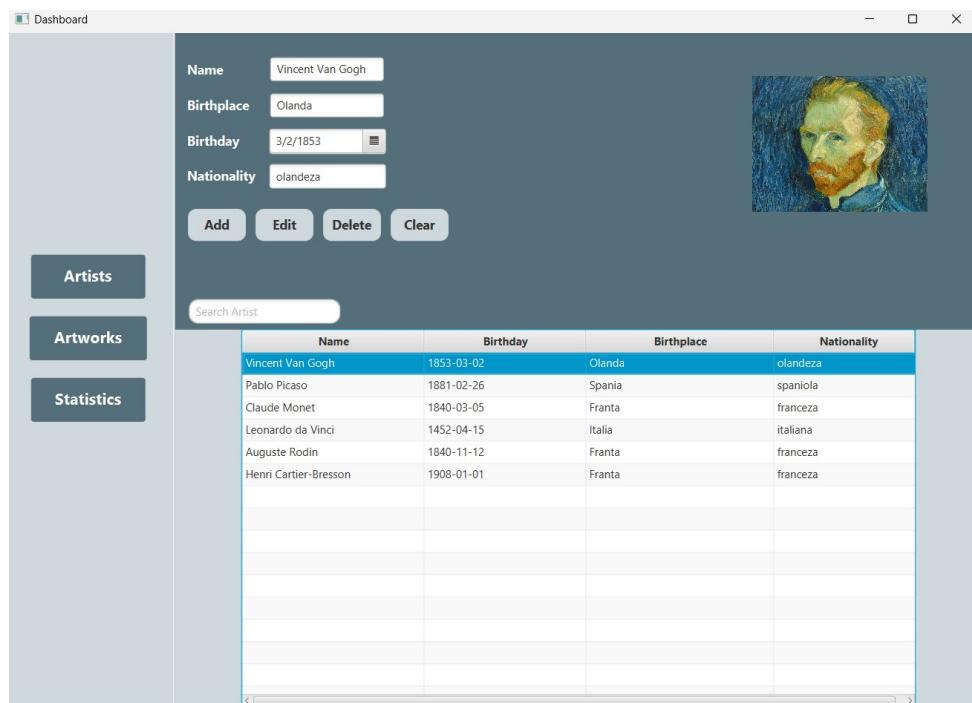


Figura 35: Interfata grafica pentru artiști

5.2 Interfata operelor de artă

Acest dashboard oferă o interfață grafică clară și funcțională dedicată administrării operelor de artă, fiind o unealtă utilă pentru pasionații de artă sau managerii de colecții. Meniul din stânga include opțiunile „Artiști” și „Opere de artă”, iar ultima secțiune este selectată. În partea superioară, un formular permite introducerea detaliilor unei opere, precum titlul, artistul, tipul, prețul și anul creației. Butoanele „ADD”, „EDIT”, „DELETE” și „ADD IMAGE” oferă flexibilitate, permitând adăugarea, modificarea sau ștergerea operelor, precum și încărcarea imaginilor corespunzătoare, vizibile în dreapta formularului.

Tabelul de mai jos prezintă o listă detaliată a operelor, cu coloane pentru titlu, artist, tip, preț și anul creației. Printre exemple se numără „Mona Lisa” de Leonardo da Vinci (86000.0, 1503), „Starry Night” de Vincent van Gogh (100000.0, 1889) și „Guernica” de Pablo Picasso (200000.0, 1937). Utilizatorii pot filtra datele după diverse criterii și pot exporta informațiile în format CSV sau DOC, datorită butoanelor dedicate. Această structură organizată și funcționalitatele suplimentare fac din dashboard un instrument practic pentru gestionarea colecțiilor de artă.

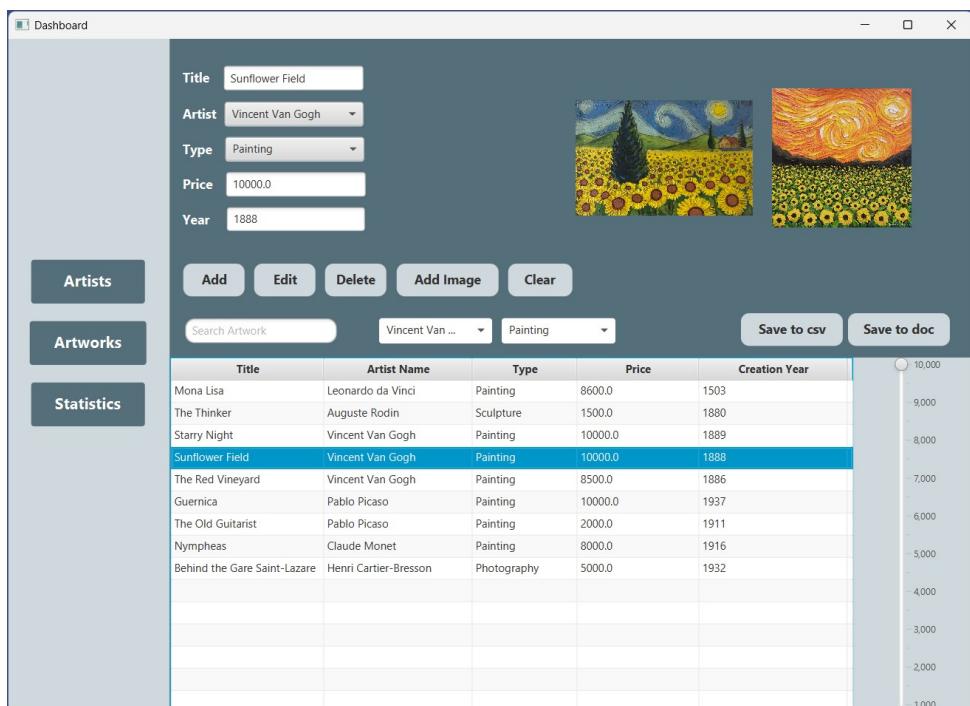


Figura 36: Interfata grafica pentru operele de arta

5.3 Interfata statisticilor

Secțiunea „Statistics” din aplicație oferă o reprezentare vizuală a datelor sub formă de grafice bară, facilitând analiza rapidă a conținutului galeriei. Graficul superior, intitulat „Type Chart”, ilustrează distribuția operelor de artă în funcție de tip (ex: pictură, fotografie, sculptură), evidențiind dominantă operelor de tip „Painting”. Graficul inferior, „Artist Chart”, afișează numărul de opere asociate fiecărui artist, permitând observarea rapidă a contribuției individuale. Această interfață permite utilizatorului să înțeleagă mai bine structura bazei de date și să identifice tendințele artistice din colecție.

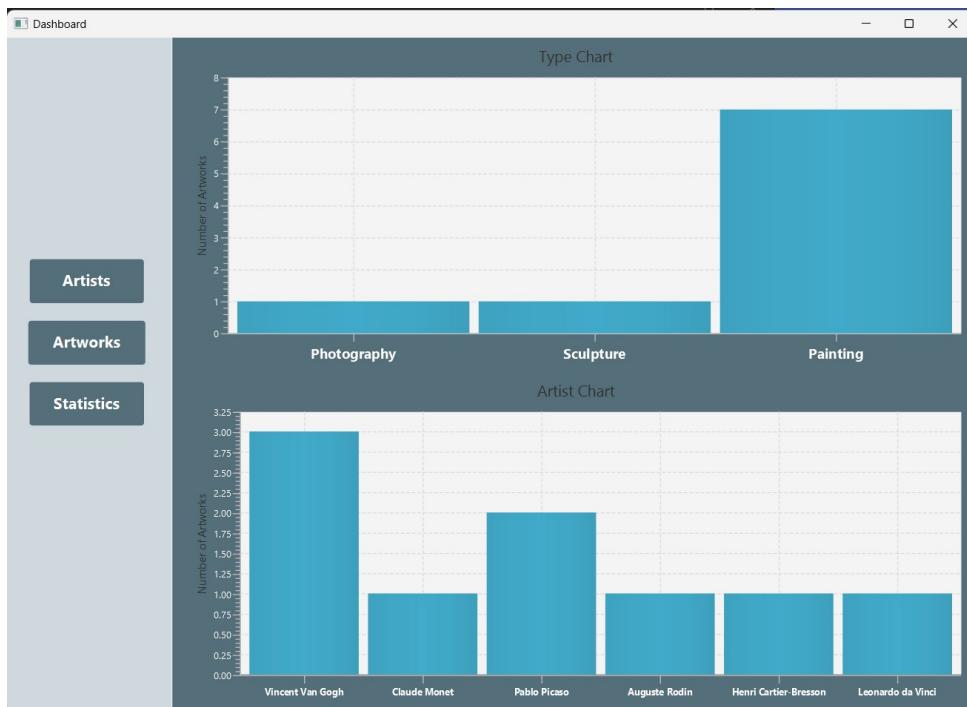


Figura 37: Interfata grafica pentru statistici