

 UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239	<i>Analiză și design pentru proiectul : Editor Grafic</i>	Pagina 1/22
	<i>Cod: T_SWDP_Analiza&Design</i>	

Versiuni:

Data	Versiune	Autor	Comentarii
07.11.2023	1.0	Boncea Natalia Georgiana Grigoraș Marius Andrei Cosmin	Aceasta este versiunea inițială care implică toate funcțiile de bază ale editorului grafic

Analiză&Design pentru Editor Grafic

Conținut:

- 1. Prezentare Generală**
- 2. Fundamente Teoretice**
- 3. Tehnologie IT**
- 4. Arhitectura și Designul Sistemului – prezentare detaliată**
- 5. Modul de operare / Prezentarea interactivității**
- 6. Portabilitate**
- 7. Update**

Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p align="center"><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <hr/> <p><i>Cod: T_SWDP_Analiza&Design</i></p>	<p align="right">Pagina 2/22</p>
--	---	--------------------------------------

1. Prezentare Generală

Aplicația propusă în cadrul acestui proiect va fi dezvoltată în C++ cu ajutorul framework-ului Qt, și va reprezenta un editor grafic complex ce integrează funcționalități atât pentru lucrul în două dimensiuni, cât și pentru mediul tridimensional. Acesta se remarcă prin similitudini notabile cu alte produse software similare precum Paint 3D, Adobe Photoshop, Adobe Illustrator, Blender, oferind o interfață intuitivă și ușor de utilizat, astfel încât utilizatorii să se adapteze rapid la mediul de lucru. Cu un set extins de instrumente pentru desen 2D și modelare 3D, această aplicație își propune să ofere o soluție cuprinzătoare pentru artiști, designeri și creatori de conținut care caută un instrument versatil și performant în domeniul editării grafice.

2. Fundamente Teoretice

Editorul grafic 3D va folosi o arhitectură bazată pe Model-View-Controller (MVC), care are următoarele componente:

- **Model** : reprezintă structurile de date de la baza aplicației; acestea conțin informații despre obiectele 2D/3D(culori, poziții etc) si starea aplicației.
 - Modelul este responsabil pentru manipularea datelor, validarea și actualizarea acestora.
- **View** : reprezintă interfața utilizator a aplicației, care permite utilizatorului să interacționeze cu modelele 2D/3D și să vadă rezultatele modificărilor în timp real.
 - Include toate elementele UI, precum ferestre, butoane, panouri de control, permite vizualizarea în timp real a proprietăților modelelor si alte componente de interfață.
- **Controller** : este componenta care gestionează logica aplicației și acționează ca intermediar între model si vizualizare.
 - Monitorizează interacțiunile utilizatorului cu interfața, precum acțiunile de editare sau manipulare a modelelor 3D.
 - Aplică schimbările asupra modelului în funcție de acțiunile utilizatorului și actualizează vizualizarea pentru a reflecta starea curentă a modelului.
 - Se asigură că datele și funcționalitatea din model și vizualizare sunt integrate corespunzător.

Pentru folosirea și înțelegerea librării OpenGL, următoarele considerații teoretice trebuie specificate :

a. Pipeline-ul de Rasterizare:

Pipeline-ul grafic reprezintă un proces sau o serie de etape prin care datele grafice sunt transformate și prelucrate pentru a fi afișate pe un ecran. În contextul bibliotecilor grafice precum OpenGL, pipeline-ul grafic este împărțit în mai multe etape distincte, fiecare cu rolul său în producerea finală a imaginii pe ecran.

b. Shader-ele:

Shader-ele sunt programe specializate care rulează pe unitățile de procesare grafică (GPU) și sunt responsabile pentru manipularea datelor grafice în timpul procesului de afișare. Aceste programe sunt esențiale în cadrul pipeline-ului grafic și permit controlul flexibil asupra aspectului final al imaginilor.

Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p align="center"><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <hr/> <p><i>Cod: T_SWDP_Analiza&Design</i></p>	<p align="right">Pagina 3/22</p>
--	---	--------------------------------------

Acestea se împart în 2 categorii :

- **Vertex Shader** : este responsabil pentru manipularea datelor de vârfuri (vertices). Prin acest shader, pot fi efectuate transformări geometrice precum translații, rotații sau scală, și se pot calcula proprietăți suplimentare, cum ar fi culorile.
- **Fragment Shader** : operează asupra fiecărui fragment individual (porțiune de pixel) generat în urma rasterizării. Aici se efectuează calculele finale ale culorii, inclusiv iluminarea, texturarea și alte efecte vizuale.

Shader-ele sunt scrise într-un limbaj de shader dedicat (cum ar fi GLSL în cazul OpenGL) și trebuie să fie compilate și legate într-un program de shader înainte de utilizare. Aceste programe sunt apoi încărcate pe GPU și executate în timpul procesului de afișare pentru a manipula și a genera imagini.

c. VBO și VAO:

VBO (Vertex Buffer Object) și VAO (Vertex Array Object) sunt două concepte cheie în OpenGL, utilizate pentru a optimiza și eficientiza manipularea datelor grafice, în special a datelor de vârfuri în cadrul unui program OpenGL.

- **Vertex Buffer Object (VBO)**: reprezintă un buffer în care sunt stocate datele de vârfuri, cum ar fi coordonatele spațiale, culorile sau alte atribute. Utilizarea VBO-urilor permite stocarea datelor pe GPU, ceea ce duce la o performanță mai bună în comparație cu transmiterea repetată a datelor de la CPU la GPU.
- **Vertex Array Object (VAO)**: reprezintă un container care păstrează configurările pentru datele de vârfuri, inclusiv referințele către VBO-uri, configurarea atributelor de vârf (precum poziția sau culoarea) și modul în care aceste date sunt organizate în memoria GPU. VAO-urile permit programatorilor să definească o singură dată configurația datelor de vârfuri și să o refolosească în mod eficient pentru diverse obiecte sau scene.

Pe lângă aceste noțiuni legate de principiul de funcționare al bibliotecii OpenGL și al întregului proces de rasterizare, este bine să înțelegem și algoritmi care stau la baza funcțiilor de procesare de imagini din aplicația noastră.

a. Conversia unei imagini color într-o imagine grayscale

Pentru a converti o imagine color într-o imagine grayscale, cele trei componente ale culorii fiecărui pixel trebuie egalizate. O metodă des folosită este medierea celor 3 componente:

$$R_{Dst} = G_{Dst} = B_{Dst} = \frac{R_{Src} + G_{Src} + B_{Src}}{3}$$

b. Reglarea opacității unei imagini

Reglarea transparenței unei imagini se poate realiza prin ajustarea canalului alfa al pixelilor. Fiecare pixel într-o imagine cu transparență are patru componente: roșu (R), verde (G), albastru (B) și alfa (A). Valoarea alfa determină nivelul de transparență, unde 0 reprezintă complet transparent, iar 255 reprezintă complet opac.

c. Reglarea balanței de culori RGB

Reglarea balanței de culoare și ajustarea cantității unei culori de bază într-o imagine pot fi realizate printr-un algoritm simplu. Un mod comun de a face acest lucru este prin ajustarea intensităților culorilor roșu (R), verde (G) și albastru (B) pentru fiecare pixel. Un exemplu de astfel de algoritm ar fi :

Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p style="text-align: center;"><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <hr/> <p><i>Cod: T_SWDP_Analiza&Design</i></p>	<p style="text-align: right;">Pagina 4/22</p>
--	--	---

red_factor = 1.2 # ajustare pentru canalul roșu
green_factor = 0.8 # ajustare pentru canalul verde
blue_factor = 1.0 # ajustare pentru canalul albastru

pentru fiecare pixel în imagine:

extrage valorile R, G, B
R, G, B = pixel

ajustează intensitățile culorilor conform factorilor
R = int(R * red_factor)
G = int(G * green_factor)
B = int(B * blue_factor)

asigură că valorile rămân în intervalul 0-255
R = max(0, min(R, 255))
G = max(0, min(G, 255))
B = max(0, min(B, 255))

actualizează pixelul cu noile valori
pixel = (R, G, B)

În acest exemplu, *red_factor*, *green_factor* și *blue_factor* sunt factorii care controlează cât de mult să fie ajustată intensitatea fiecărui canal de culoare. Prin modificarea valorilor acestor factori se obțin efectele dorite în reglarea balanței de culoare și în modificarea cantității de culoare în imagine.

3. Tehnologie IT

Limbajul de programare C++ este un limbaj de programare de nivel înalt, cu scop general, o extensie a limbajului C care adaugă funcționalități specifice programării orientate pe obiect (OOP), având ca și caracteristici cheie ale programării orientate pe obiect în acest context:

1. **Programare Orientată pe Obiect (OOP):** C++ permite definirea și utilizarea claselor, obiectelor și conceptelor OOP. Clasa reprezintă o structură care poate să includă date (variabile de membru) și funcții (metode). Obiectele sunt instanțe ale claselor, și acestea pot comunica între ele prin intermediul mesajelor.
2. **Encapsulare:** C++ oferă encapsulare, care permite împachetarea datelor și a funcțiilor asociate într-o singură entitate (clasă). Accesul la datele și funcțiile unei clase poate fi controlat prin intermediul specificatorilor de acces (public, private, protected), oferind astfel un grad crescut de securitate și modularitate.
3. **Polimorfism:** C++ susține polimorfismul, care poate fi static (compile-time) sau dinamic (runtime). Polimorfismul permite crearea de funcții sau clase care pot fi utilizate în mai multe moduri, în funcție de context.
4. **Moștenire:** O altă caracteristică importantă a limbajului C++ este moștenirea, care permite o clasă să preia caracteristicile unei alte clase. Aceasta facilitează reutilizarea codului și definirea unor ierarhii de clase.

Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p align="center"><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <hr/> <p><i>Cod: T_SWDP_Analiza&Design</i></p>	<p align="right">Pagina 5/22</p>
--	---	--------------------------------------

5. **Alocare dinamică de memorie:** C++ permite alocarea și dealocarea dinamică a memoriei prin intermediul operatorilor new și delete. Aceasta oferă flexibilitate în gestionarea resurselor în timpul rulării programului.

Aceste caracteristici reprezintă fundamentul programării orientate pe obiect și vor fi frecvent utilizate în cadrul proiectului.

Qt este un **framework** de dezvoltare software puternic și extensibil, folosit pentru crearea de aplicații cu interfețe grafice (GUI), dar nu numai. Deținut de compania The Qt Company, framework-ul este scris în C++, dar oferă suport și pentru alte limbaje de programare, precum Python.

Qt este cunoscut pentru portabilitatea sa. Aplicațiile dezvoltate cu Qt pot fi compilate și rulate pe diverse platforme, inclusiv Windows, Linux, macOS, Android și iOS, fără a necesita modificări semnificative ale codului sursă. În contextul acestui proiect, platforma va fi Windows.

Avantajele aduse de acest framework sunt numeroase, dar cele care ies în evidență sunt:

- **Interfață Utilizator Consistentă:** Biblioteca Qt oferă componente grafice (widget-uri) pentru construirea interfețelor utilizator, asigurând o aparență și comportament consistente pe toate platformele. Acest aspect este crucial pentru un editor grafic, unde experiența utilizatorului și familiaritatea cu interfața sunt esențiale.
- **Gestionarea Evenimentelor și Semnale/Slot-uri:** Mecanismul de gestionare a evenimentelor și semnale/slot-uri din Qt facilitează interacțiunea dintre diferite componente ale editorului grafic. Acest lucru permite o gestionare mai ușoară a acțiunilor utilizatorului și actualizarea corespunzătoare a interfeței grafice.
- **Extensibilitate și Modularitate:** Qt este cunoscut pentru arhitectura sa modulară, permițând dezvoltatorilor să utilizeze doar componentele necesare proiectului lor. Această caracteristică contribuie la menținerea unui cod curat și ușor de gestionat.
- **O gamă largă de biblioteci esențiale :** *Qt Core* - oferă funcționalități de bază, cum ar fi gestionarea datelor, manipularea fișierelor și directoarelor, *Qt Widgets* - conține componente grafice (widget-uri) pentru construirea interfețelor utilizator, inclusiv butoane, etichete, casete de text, ferestre, meniuri etc, *Qt Gui* - furnizează funcționalități pentru manipularea graficii, inclusiv desenarea primitivelor grafice, gestionarea culorilor și fonturilor, și suport pentru interacțiunea cu dispozitivele de intrare, *Qt OpenGL* - furnizează facilități pentru integrarea graficii 3D și manipularea directă a contextului OpenGL în aplicații Qt.
- **Comunitate Activă și Documentație Bogată :** Fiind o tehnologie populară, Qt beneficiază de o comunitate activă și de o documentație extinsă. Acest lucru este de neprețuit în procesul de dezvoltare, oferind suport și resurse utile.

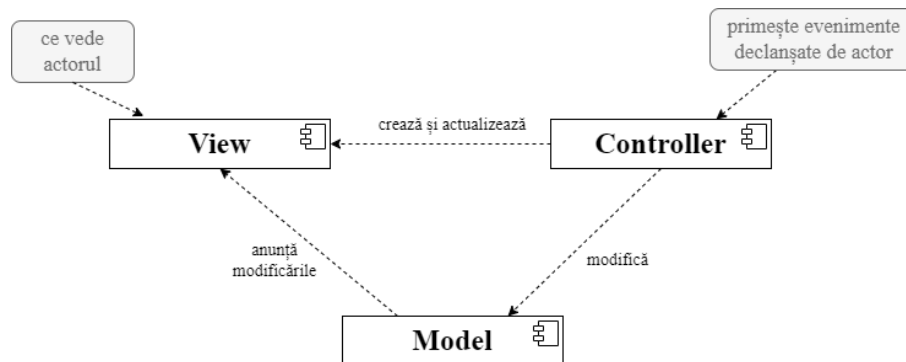
OpenGL (Open Graphics Library) este o specificație și o interfață de programare pentru dezvoltarea aplicațiilor grafice, cu un accent deosebit pe redarea 2D și 3D. Acesta oferă funcționalități puternice pentru redarea graficii în două și trei dimensiuni. Este folosit într-o varietate de aplicații, inclusiv jocuri video, simulări, programe CAD, vizualizări științifice și multe altele.

Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

4. Arhitectura si Designul Sistemului – detaliile

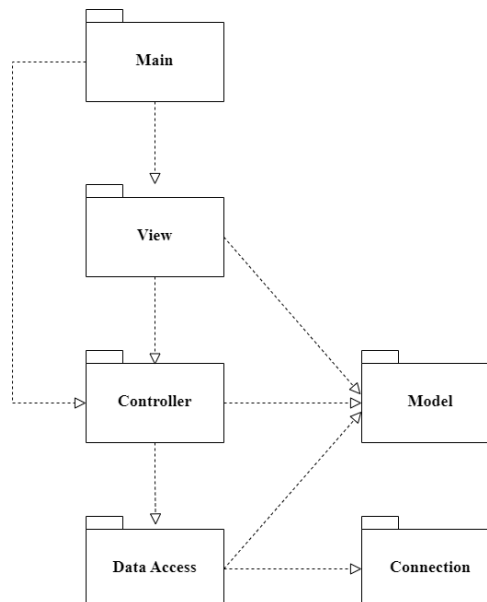
4.1. Diagrama generala a sistemului

Așa cum am precizat și în partea de fundamente teoretice, arhitectura sistemului nostru este una de tipul Model-View-Controller. Mai jos este prezentată diagrama arhitecturii implementate și ce relații există între componentele care o alcătuiesc:



4.2. Diagrame detaliate la nivel de modul

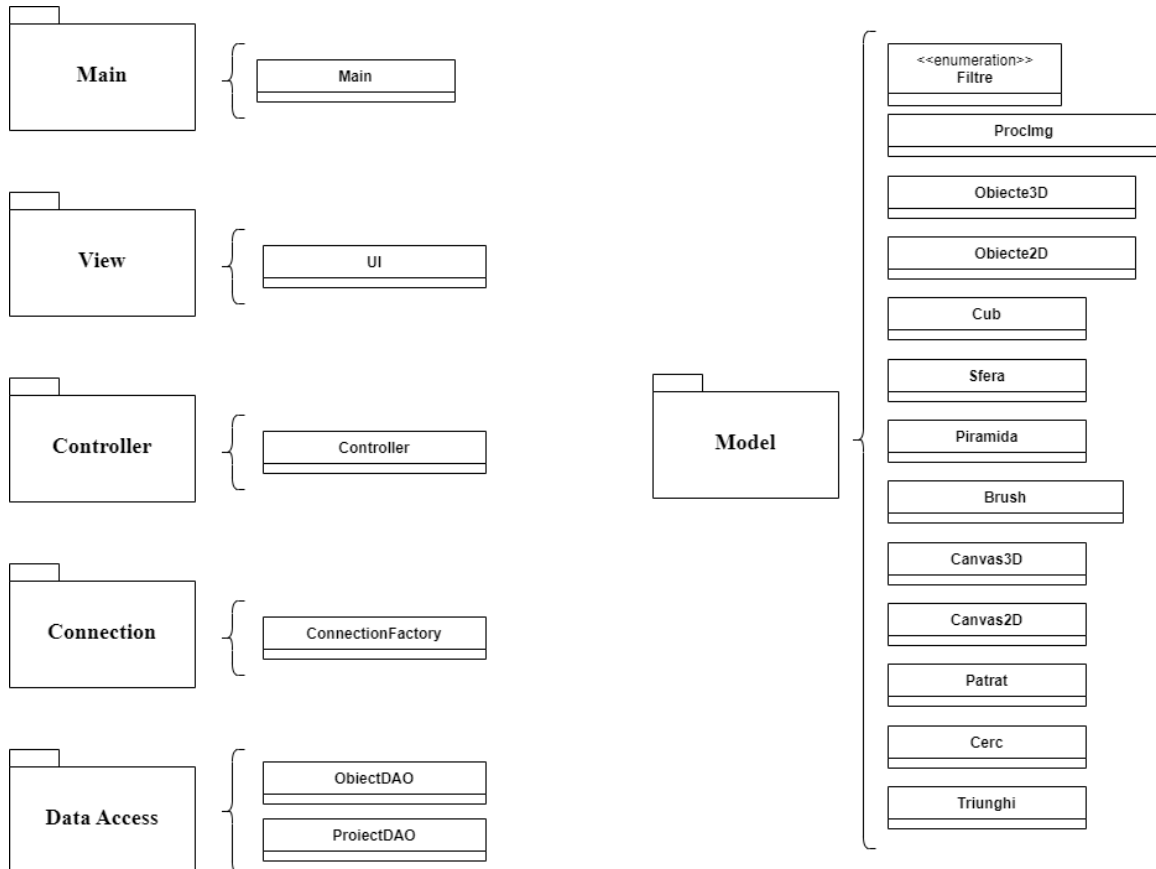
Fiecare modul din diagrama generală a sistemului poate fi materializat sub forma unui pachet de clase. Așadar, diagrama de pachete rezultată arată astfel :



Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p align="center"><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <hr/> <p><i>Cod: T_SWDP_Analiza&Design</i></p>	<p align="right">Pagina 7/22</p>
--	--	--------------------------------------

Următoarea diagramă detaliată suprinde conținutul fiecărui modul :

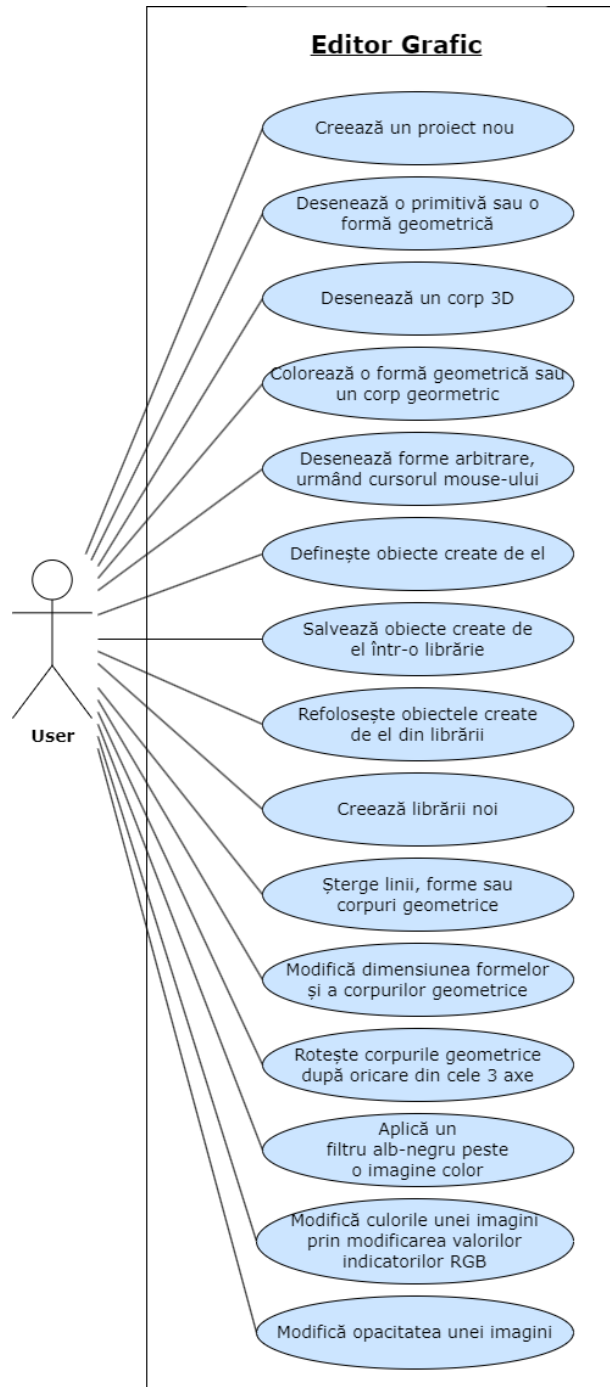


4.3. Cazuri de utilizare

În continuare este prezentată diagrama use-case, împreună cu un exemplu de caz de utilizare a aplicației de editare grafică 3D pentru desenarea unei case 3D:

Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p align="center"><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <hr/> <p>Cod: T_SWDP_Analiza&Design</p>	<p align="center">Pagina 8/22</p>
--	---	---------------------------------------



Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p align="center"><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <hr/> <p><i>Cod: T_SWDP_Analiza&Design</i></p>	<p align="center">Pagina 9/22</p>
--	--	---------------------------------------

Caz de utilizare: desenarea unei case în 3D

Actorul principal: utilizatorul

Scop: scopul actorului este de a crea un nou proiect în care va desena o casă în 3D, iar apoi o va salva într-o bibliotecă pentru a o putea reutiliza în viitor.

Scenariul ideal de succes:

Precondiții: Aplicația va afișa o fereastră în care autorul trebuie să ofere specificațiile necesare pentru realizarea proiectului nou.

Pași:

1. Utilizatorul deschide aplicația și creează un nou proiect
2. Utilizatorul inserează un cub
3. Utilizatorul inserează o piramidă
4. Utilizatorul rotește corpurile după axa Y până ajunge la unghiul dorit
5. Utilizatorul realizează ușa și geamurile prin inserarea unor pătrate
6. Utilizatorul selectează una din laturile unui pătrat și trage de ea pentru a-l deforma și a-l transforma în dreptunghi
7. Utilizatorul selectează una dintre forme și apasă butonul de colorare
8. Aplicația deschide o fereastră din care utilizatorul își poate alege o culoare standard sau poate alege o altă culoare prin introducerea unor valori corespunzătoare codului RGB
9. Obiectul selectat se colorează
10. Utilizatorul repetă pașii 7-9 până când toate obiectele sunt colorate
11. După ce desenul este finalizat, utilizatorul creează o nouă librărie de obiecte desenate
12. Utilizatorul salvează obiectul desenat în librăria proaspăt creată pentru a fi reutilizat în viitor
13. Utilizatorul închide aplicația.

Postcondiții: În urma închiderii aplicației provocată de utilizator, nu mai are loc nicio schimbare de stare în cadrul sistemului.

Scenarii alternative:

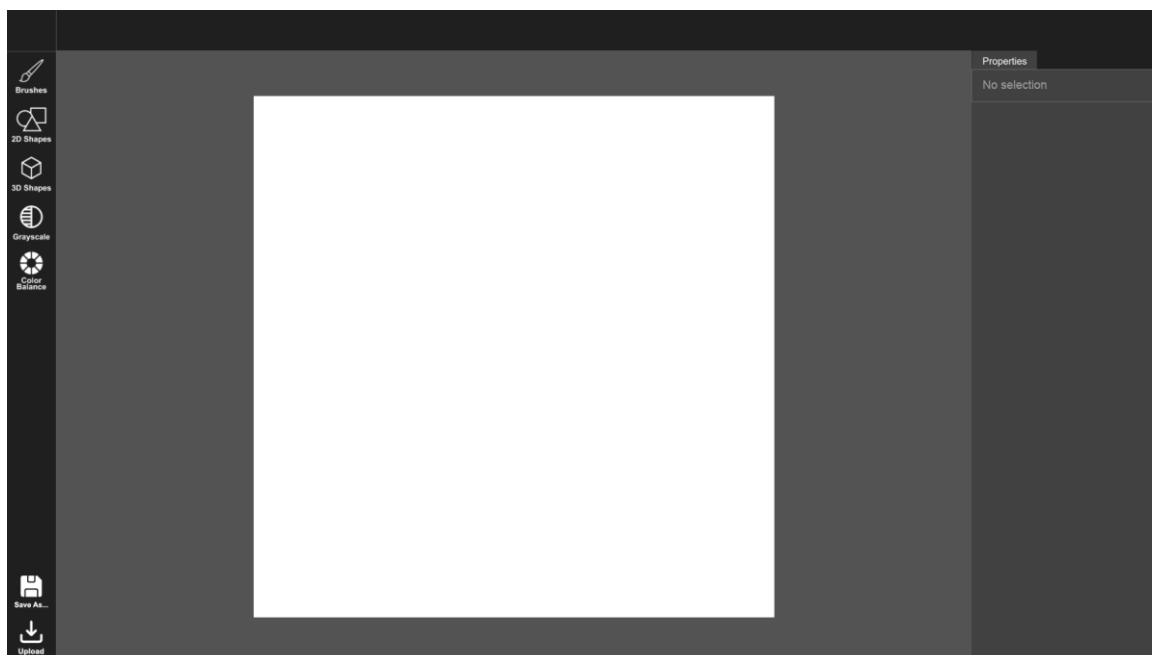
- Utilizatorul încearcă să șteargă un obiect dintr-o librărie
 - Se deschide o fereastră în care utilizatorul este atenționat că obiectul respectiv nu va mai putea fi recuperat și este întrebat dacă vrea să continue acțiunea;
 - Dacă vrea să șteargă obiectul, atunci va apăsa pe butonul “Da”, iar fereastra se va închide și obiectul va fi șters;

Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p align="center"><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <hr/> <p><i>Cod: T_SWDP_Analiza&Design</i></p>	<p align="center">Pagina 10/22</p>
--	--	--

- Dacă nu vrea să ștergă obiectul, atunci va apăsa pe butonul “Nu”, iar fereastra se va închide și obiectul va rămâne în librărie.
- Utilizatorul salvează un obiect cu un nume deja existent:
 - Se deschide o fereastră în care utilizatorul este atenționat că există deja în librărie un obiect cu același nume și are de ales între 3 opțiuni: să înlocuiască obiectul vechi cu cel nou, să modifice numele obiectului nou sau să anuleze salvarea obiectului;
 - Dacă utilizatorul alege să înlocuiască obiectul vechi cu cel nou, obiectul vechi cu același nume este șters și în locul lui este salvat cel nou;
 - Dacă utilizatorul alege să modifice numele obiectului nou, fereastra se va închide și utilizatorul va reveni în punctul de denumire a obiectului;
 - Dacă utilizatorul alege să anuleze salvarea obiectului, fereastra se va închide și utilizatorul va reveni la planșa de desen, fără a se salva obiectul creat.

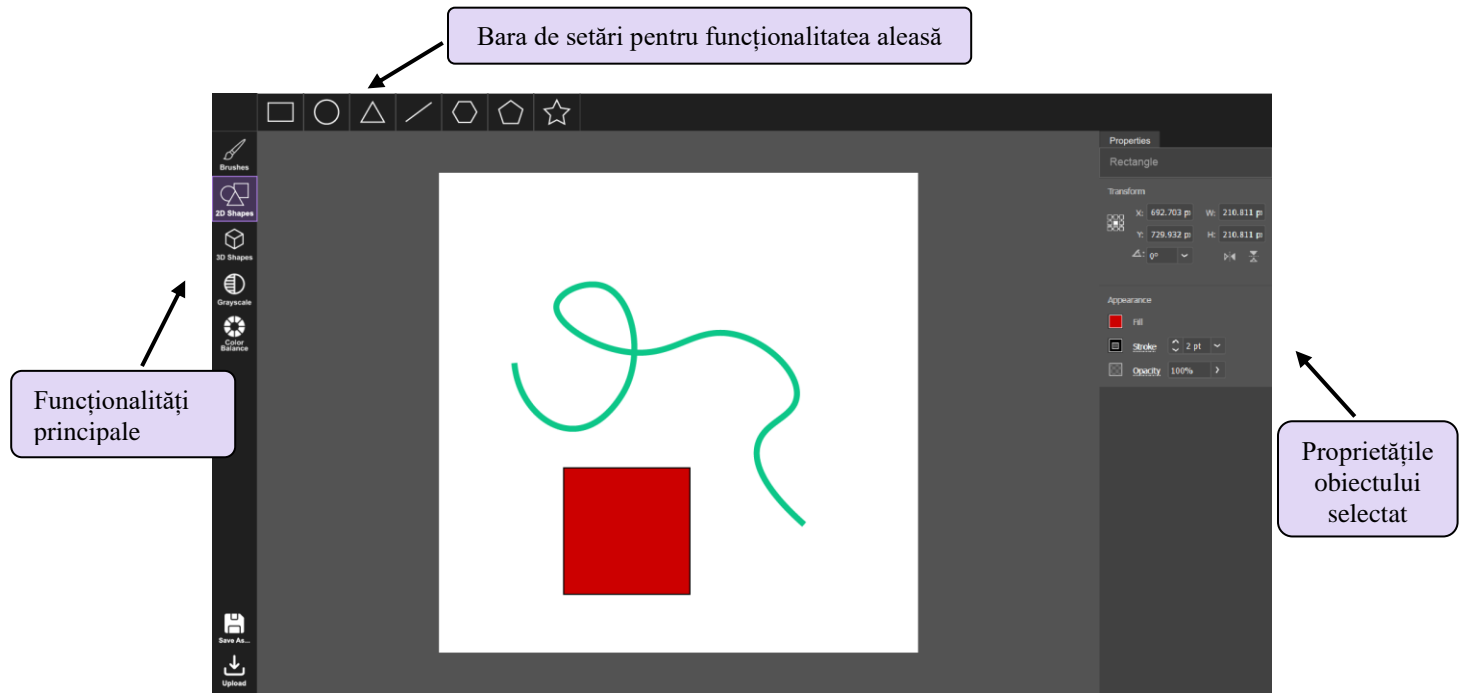
4.4. Designul interfeței utilizator



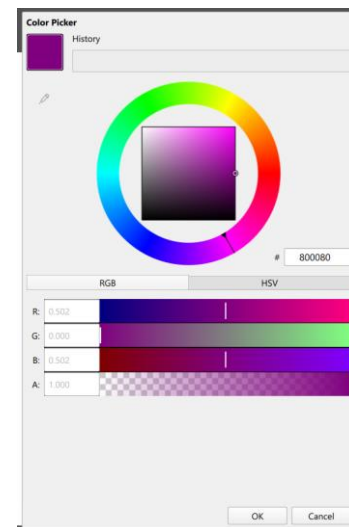
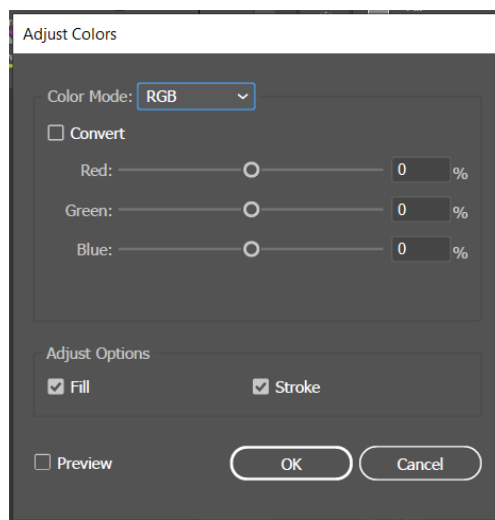
Interfața principală a aplicației

Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <p>Cod: T_SWDP_Analiza&Design</p>	<p>Pagina 11/22</p>
--	--	-------------------------



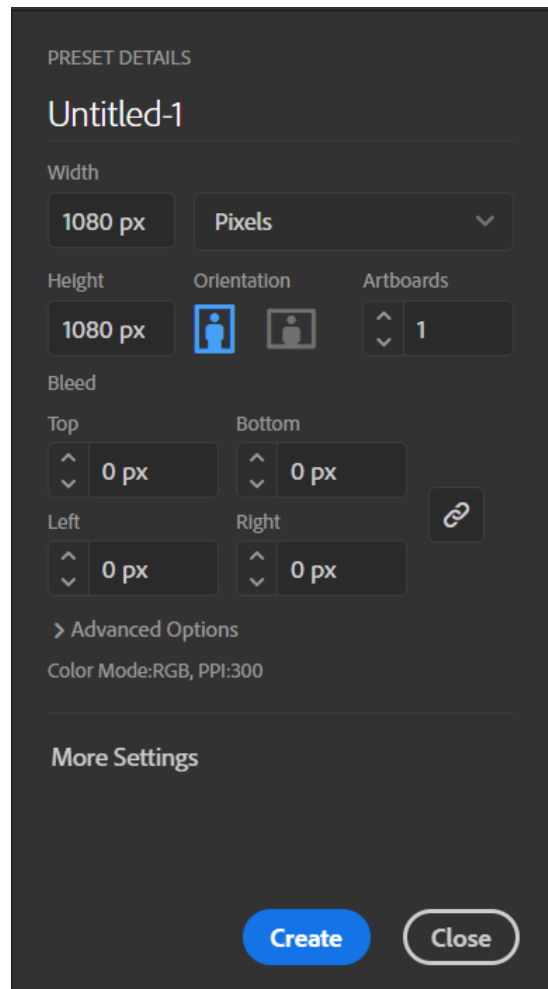
Interfața grafică a aplicației – Localizare funcționalități



Ferestre pentru setări (Color Balance și Color Picker)

<p>Recomandat: (data, semnatura)</p>	<p>Aprobat: (data, semnatura)</p>
<p>Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania</p>	<p>Telefon: Fax: E-mail: Web:</p>

 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <p><i>Cod: T_SWDP_Analiza&Design</i></p>	<p>Pagina 12/22</p>
--	---	-------------------------



Fereastra de setări pentru crearea unui nou proiect

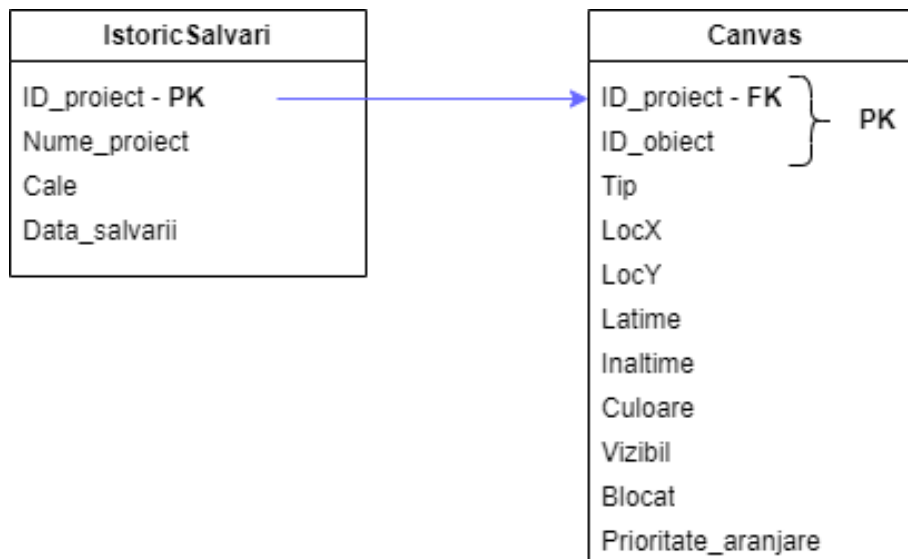
Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

 UNIVERSITATEA TEHNICĂ <small>DIN CLUJ-NAPOCA</small> Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239	<p align="center"><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <hr/> <p><i>Cod: T_SWDP_Analiza&Design</i></p>	Pagina 13/22
---	---	-----------------

4.5. Structura bazei de date

Această aplicație de editare grafică va accesa o bază de date cu două tabele. Primul tabel va reține date despre ultimele 10 proiecte salvate în calculator, precum numele proiectului, calea prin directoare către fiecare proiect și data ultimei salvări. Acest tabel ne va oferi posibilitatea de a accesa unul dintre primele 10 cele mai recente proiecte modificate direct din aplicație, fără să îl căutăm noi prin toate directoarele.

Cel de-al doilea tabel ne va ajuta să facem o listă de obiecte inserate în proiect și conține atribute precum locația obiectului în planșă, tipul acestuia, dimensiunea chenarului de încadrare, culoare, dacă poate fi modificat, dacă este vizibil și prioritatea lui la aranjarea în proiect. Această listă va ușura selectarea sau ștergerea unui obiect din planșă și va crește lizibilitatea proiectului.



4.6. Design OO

Întrucât aplicația noastră este una ce aplică principiile programării orientate pe obiecte, structura acesteia este divizată în mai multe clase. Pentru o prezentare sumară a relațiilor dintre aceste clase am realizat o diagramă UML de clase, urmând ca în partea de implementare să se ofere o prezentare mai detaliată a atributelor și a funcționalităților pe care le furnizează.

Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

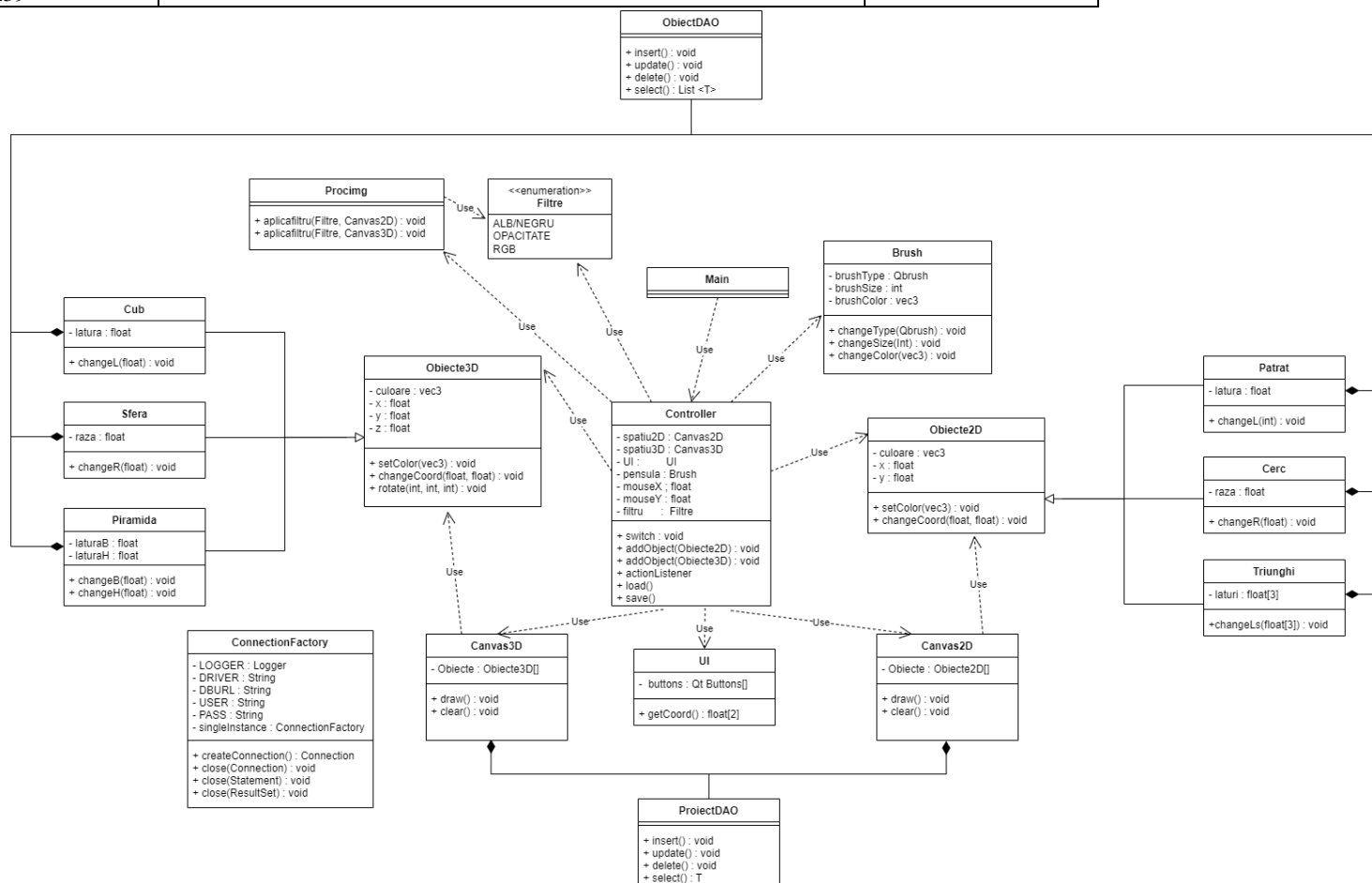


**UNIVERSITATEA
TEHNICĂ**
DIN CLUJ-NAPOCA
Nume: BONCEA Natalia
Georgiana
GRIGORAȘ Marius
Andrei Cosmin
Grupa: 30239

Analiză și design pentru proiectul : Editor Grafic

Pagina
14/22

Cod: T_SWDP_Analiza&Design



Recomandat:

(data, semnatura)

Compania/Numele Universitatii

Adresa:
3400 Cluj-Napoca
Romania

Aprobat:

(data, semnatura)

Telefon:

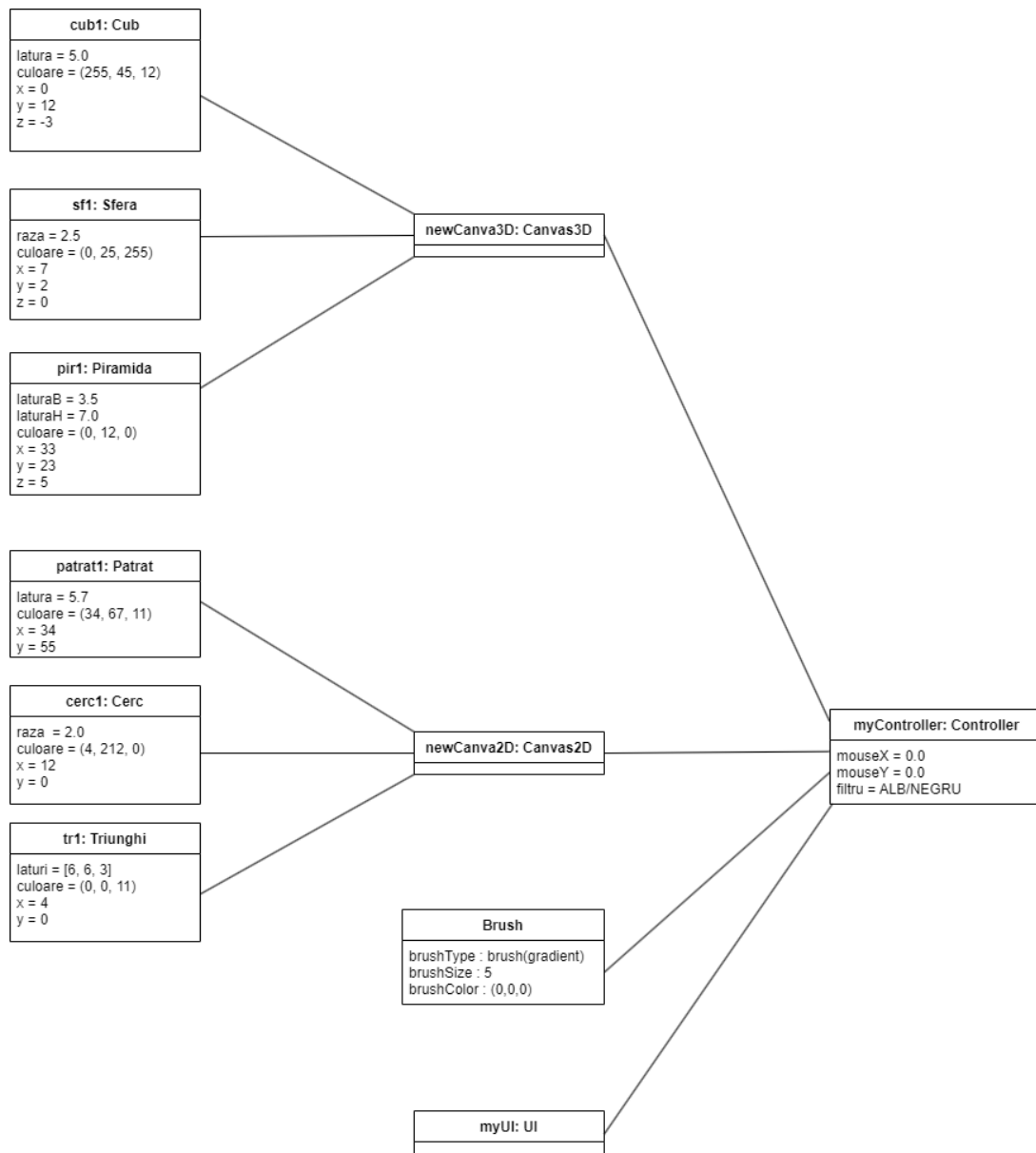
Fax:

E-mail:

Web:

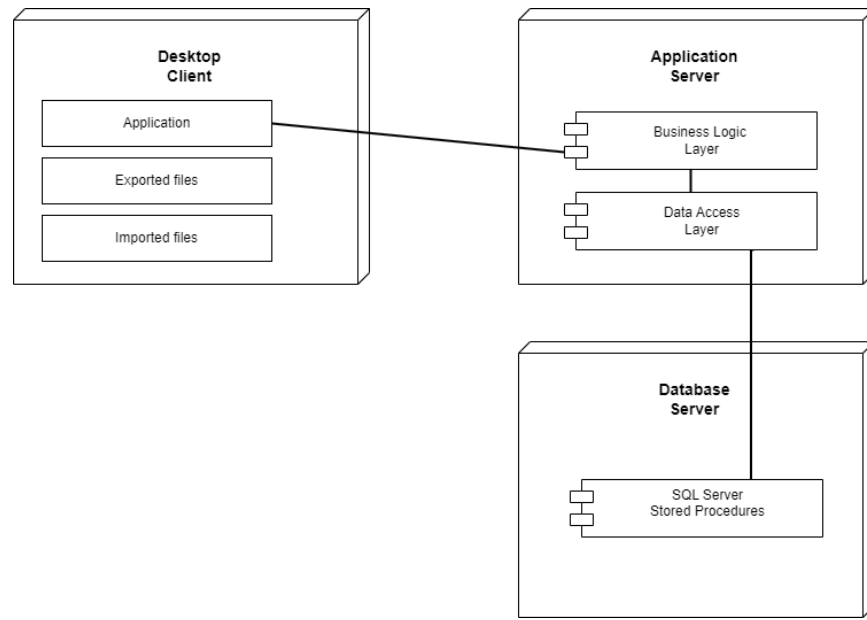
 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p align="center"><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <hr/> <p><i>Cod: T_SWDP_Analiza&Design</i></p>	<p align="center">Pagina 15/22</p>
--	---	--

Odată cu rularea și utilizarea aplicației de către un actor, se creează mai multe obiecte, care depind unele de altele. Diagrama de mai jos are ca scop exemplificarea unor obiecte care pot fi create pe parcursul rulării programului și cum depind unele de altele :

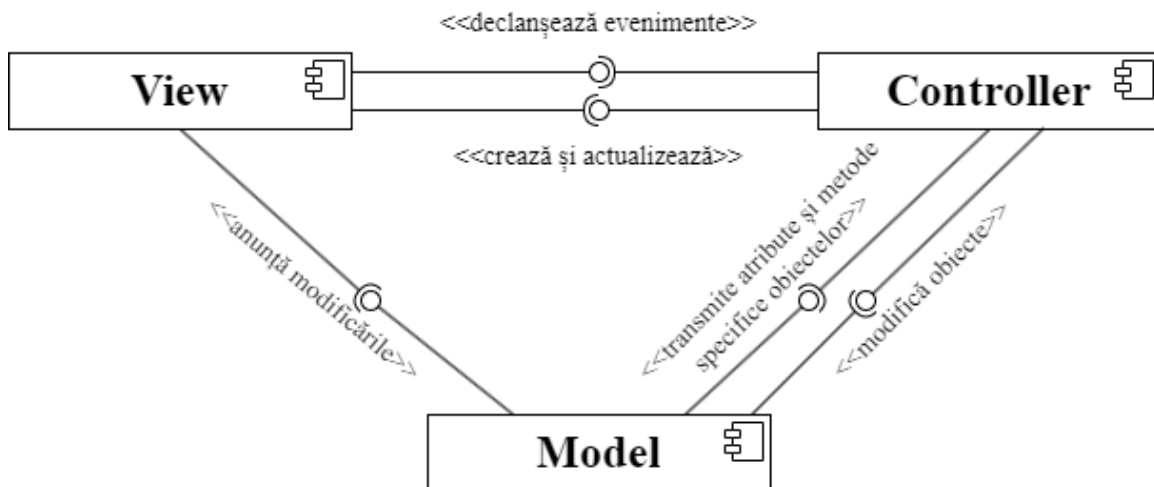


Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

4.7. Diagrama de desfășurare

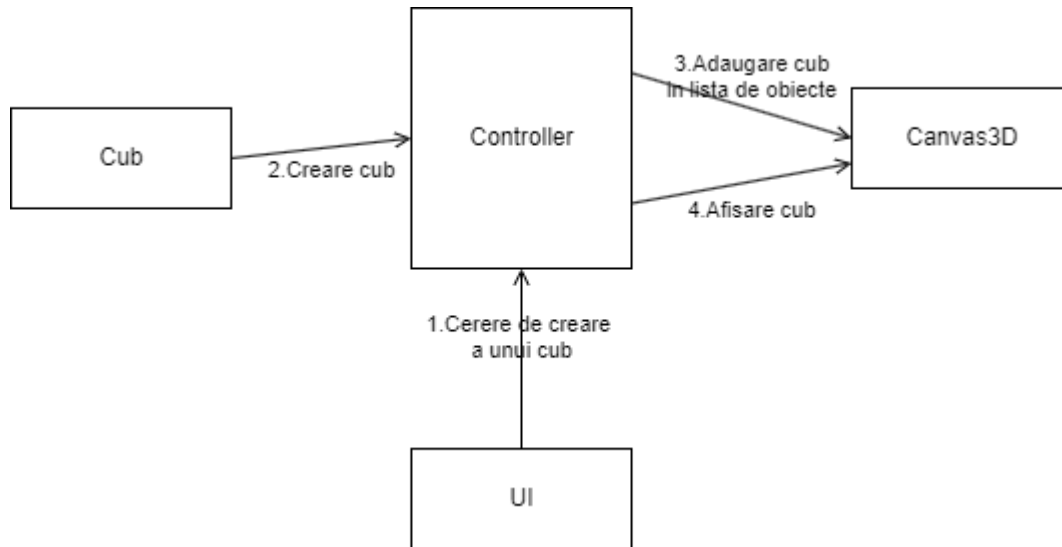


4.8. Diagrama de componente

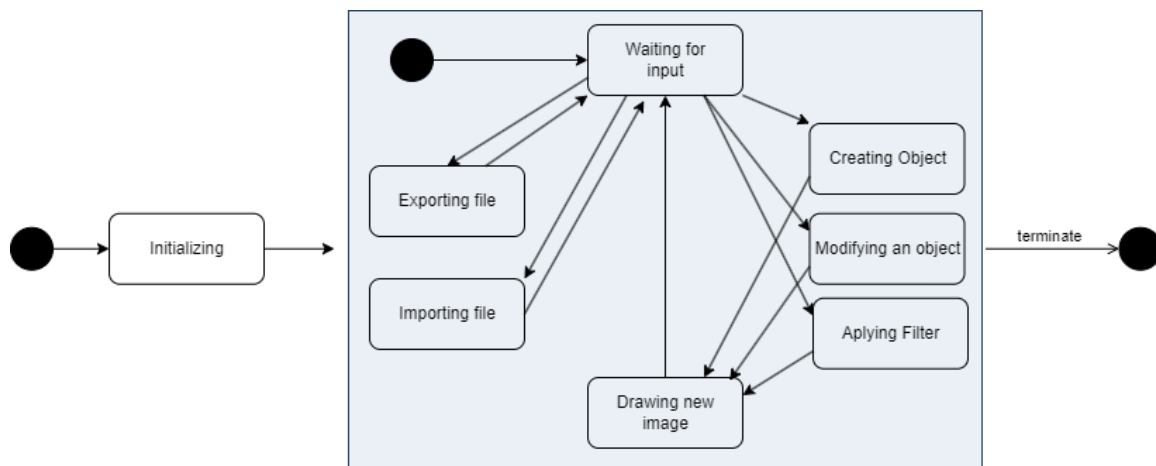


Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

4.9. Diagrama de comunicare



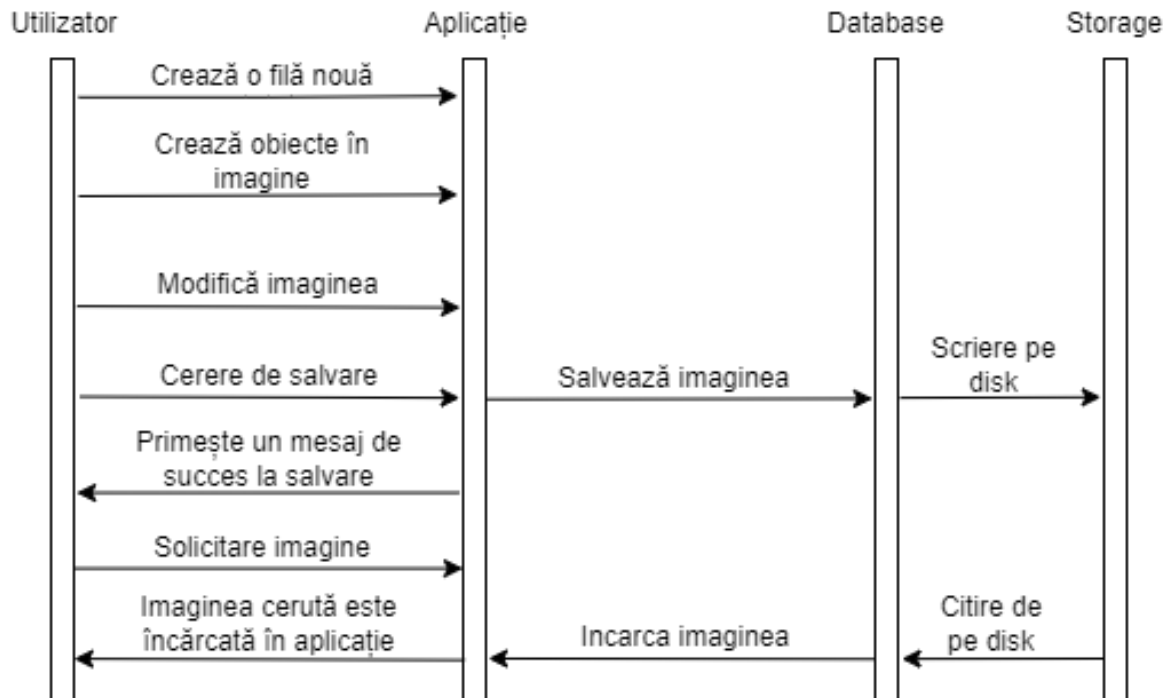
4.10. Diagrama de activitate



Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <hr/> <p><i>Cod: T_SWDP_Analiza&Design</i></p>	<p>Pagina 18/22</p>
--	---	-------------------------

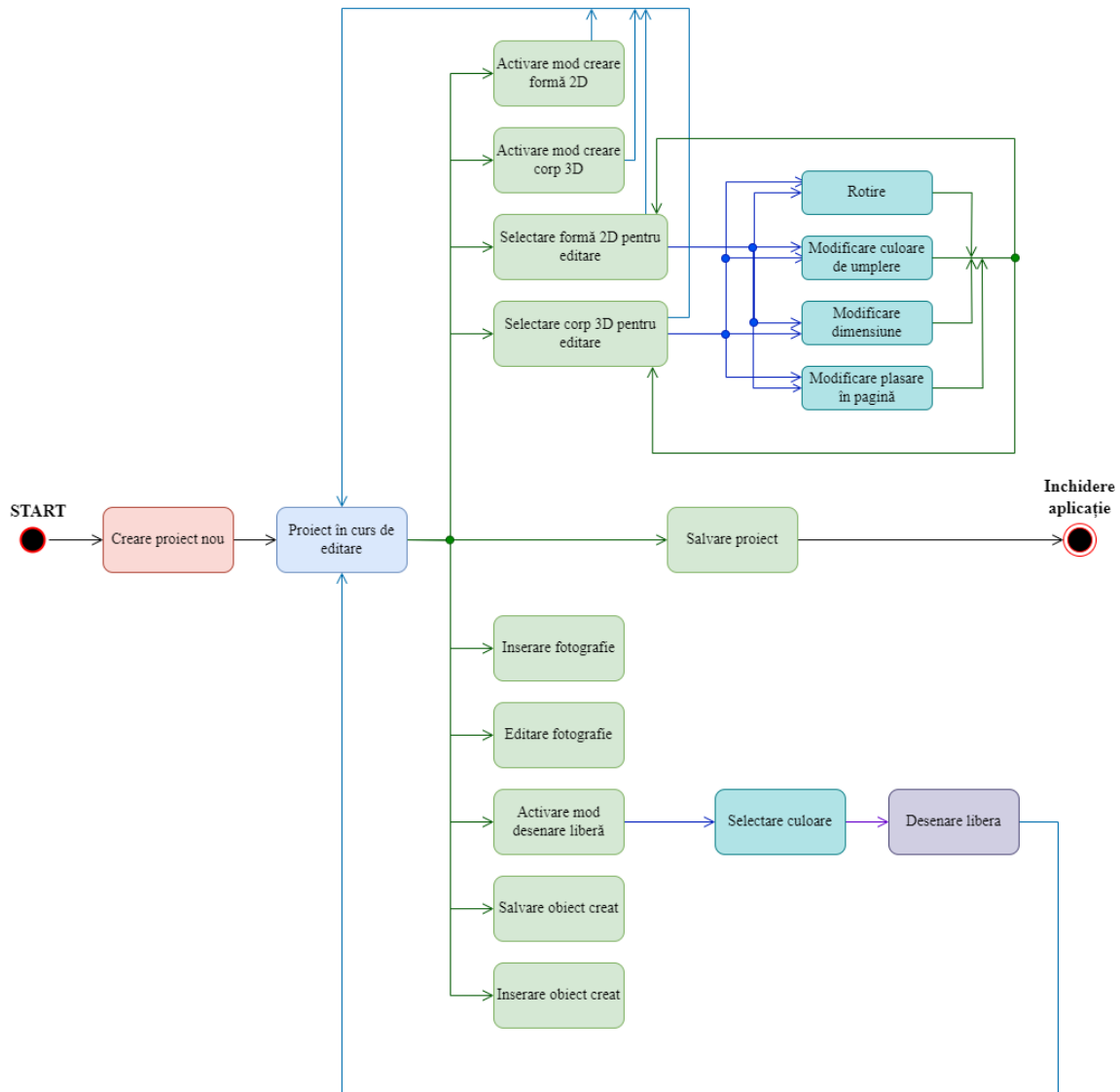
4.11. Diagrama de secvențe



Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <p>Cod: T_SWDP_Analiza&Design</p>	<p>Pagina 19/22</p>
--	--	-------------------------

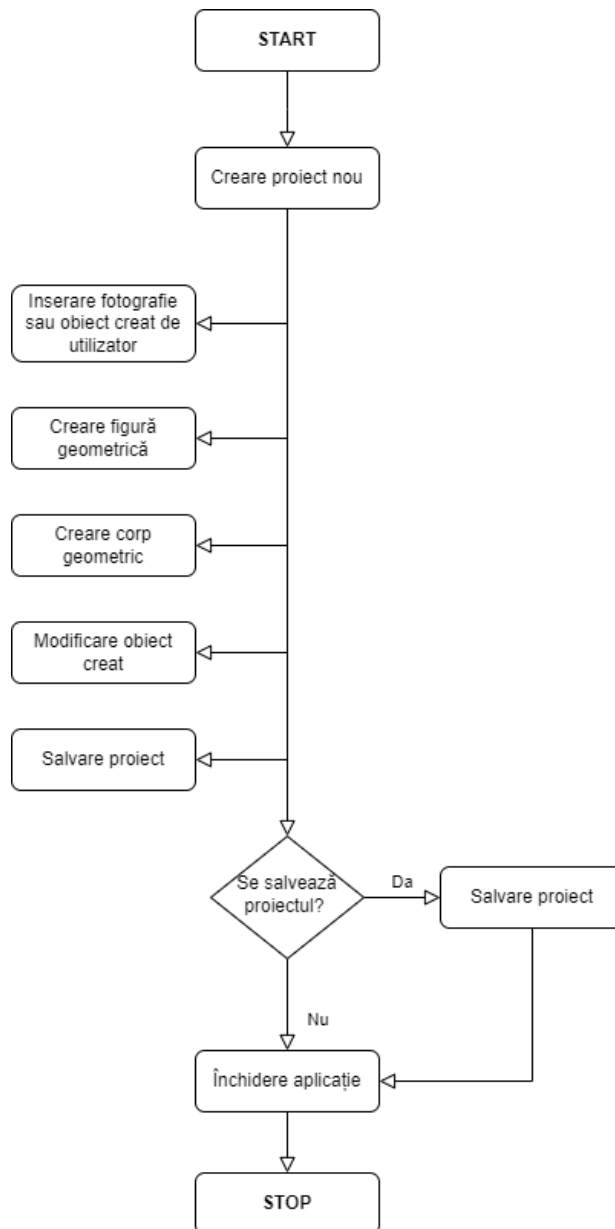
4.12. Diagrama de tranziție de stări



Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <p><i>Cod: T_SWDP_Analiza&Design</i></p>	<p>Pagina 20/22</p>
--	---	-------------------------

4.13. Schema logică



Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

 <p>UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</p>	<p align="center"><i>Analiză și design pentru proiectul : Editor Grafic</i></p> <hr/> <p>Cod: T_SWDP_Analiza&Design</p>	<p align="right">Pagina 21/22</p>
--	---	---------------------------------------

5. Modul de operare/ Prezentarea interactivității

La deschiderea aplicației, utilizatorul va fi întâmpinat de fereastra de creare a unui nou proiect. Aceasta va trebui să completeze câmpurile necesare cu particularitățile pe care și le dorește la crearea planșei de desen, ca de exemplu dimensiunile acesteia, orientarea (landscape sau portret) și numele proiectului. După ce a completat toate câmpurile necesare, utilizatorul va apăsa butonul “Create”.

Pe ecran va apărea fereastra principală a aplicației, așa cum este reprezentată și în imaginea din capitolul anterior. Fereastra este delimitată în 4 zone:

- Zona canvas – zona efectivă pentru desenat și editat, în centrul ferestrei
- Setul de butoane din stânga – pun la dispoziție funcționalitățile principale ale aplicației (pensula de desenat, adăugare de corpuri și forme geometrice, opțiuni de editare a pozelor). Chiar în partea de jos, se pot observa cele două butoane pentru salvarea proiectului, respectiv încărcarea unui obiect deja existent sau a unei fotografii în proiect
- Bara de setări și opțiuni din partea de sus - aici vor fi afișate opțiunile și setările necesare la crearea unui nou obiect (setul de forme geometrice pe care le putem crea, culoarea pensulei pentru desenat, grosimea liniei etc.)
- Bara de proprietăți din dreapta – dacă selectăm un obiect, în această zonă se vor afișa proprietățile actuale ale obiectului. Tot de aici, putem aduce modificări obiectului.

Urmează procesul de desenare și editare. Utilizatorul poate încărca în proiect fotografii sau obiecte create de el prin apăsarea butonului de “Upload”. După aceea, el va selecta oricare din butoanele din stânga pentru a crea obiecte noi, pentru a desena cu mouse-ul sau pentru a edita fotografia. La apăsarea unui buton, el va trebui să facă setările și alegerile necesare din bara de sus sau din fereastra care va apărea pe ecran. Dacă dorește să modifice obiecte deja existente pe planșă, utilizatorul va selecta obiectul respectiv dând click pe el și va aduce modificările necesare din bara de proprietăți din dreapta.

În cele din urmă, dacă utilizatorul dorește să salveze opera creată, va selecta butonul “Save As...”, care îi va deschide o fereastră cu aplicația de gestionare a fișierelor folosită în mod frecvent, de unde poate schimba denumirea proiectului, formatul de salvare și poate alege fișierul în care să se salveze proiectul.

Când utilizatorul va dori să înceteze activitatea în cadrul aplicației, poate pur și simplu să închidă aplicație. De menționat, însă, că dacă proiectul nu a fost salvat înainte de părăsirea aplicației, pe ecran va apărea o casetă de dialog, în care utilizatorul va trebui să aleagă dacă vrea să salveze proiectul înainte sau nu.

6. Portabilitate

Qt furnizează un API consistent pentru dezvoltarea aplicațiilor, indiferent de platforma pe care rulează. Aceasta înseamnă că funcționalitățile specifice ale sistemului de operare sunt abstractizate printr-un set comun de interfețe, facilitând portabilitatea și OpenGL este proiectat pentru a fi independent de platformă, ceea ce rezultă că din nou codul scris pentru o platformă poate fi portat relativ ușor pe alte platforme fără a necesita modificări semnificative

Însă, pentru acest proiect, unica platforma va fi Windows cu posibilitatea ca pe viitor să realizăm versiuni puțin modificate pentru platforme diferite.

Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web:

<div> UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA Nume: BONCEA Natalia Georgiana GRIGORAȘ Marius Andrei Cosmin Grupa: 30239</div>	<i>Analiză și design pentru proiectul : Editor Grafic</i>	Pagina 22/22
	<i>Cod: T_SWDP_Analiza&Design</i>	

7. Update

7.1. Testare și validare

Testarea efectuată a verifica faptul că aplicația noastră funcționează corespunzător se poate clasifica în două categorii: testare de unități și testare de sistem.

Pentru testarea de unități, am luat fiecare clasă în parte și am verificat dacă se comportă așa cum ne-am așteptat. Această testare a presupus, la rândul ei, două etape distincte, în funcție de obiectul supus testării: verificarea componentelor pentru interfața grafică cu utilizatorul, respectiv verificarea funcționalităților aplicației.

După obținerea unor rezultate promițătoare, am combinat cele două părți în sistemul principal. Aici intervine a doua categorie de testare, și anume testarea de sistem, care a presupus verificarea funcționării corecte a aplicației în diverse situații și primind diverse input-uri de la utilizator.

În urma acestor testări, am descoperit mai multe bug-uri, pe care am reușit parțial să le ameliorăm. Pentru o evidență mai bună a acestora, vom surprinde o parte dintre bug-urile aplicației noastre până la momentul actual în tabelul următor.

7.2. Bug list

Crt. No	Testing Scenario	Bug/Bugs	Bug Identification Date	Bug Fixing Date	Status [Open/Closed]
1	Lansarea aplicatie	Camera nu funcționează corect la lansarea aplicației	17.01.2024	18.01.2024	Closed
2	Mișcarea camerei prin obiecte	Interfața devine iluminată	16.01.2024	17.01.2024	Closed
3	Selectarea unui obiect care acoperă tot ecranul	Nu se mai poate deselecta un obiect din cauza dimensiunii acestuia	18.01.2024	-----	Open
4	Apăsarea cu mouse-ul pe obiectul importat	Obiectul importat nu poate fi selectat	16.01.2024	-----	Open
5	Încărcarea unor corpuri și formelor predefinite	La lansarea aplicației, corpurile 3D și formele 2D predefinite adăugate nu sunt vizibile până la importarea unui obiect	16.01.2024	18.01.2024	Closed
6	Utilizarea brush-ului	Brush-ul nu funcționează în mediul 3D	17.02.2024	-----	Open

Recomandat: (data, semnatura)	Aprobat: (data, semnatura)
Compania/Numele Universitatii Adresa: 3400 Cluj-Napoca Romania	Telefon: Fax: E-mail: Web: