Editor de imagini

Facultatea de Automatică și Calculatoare

Studenti: *Vezendan Alexandra, Petrea Irina*

An 3, grupa 30236

. . . . . . . . . . . . . .

Cuprins:

1. **Specificatii si analiza sistemului**
2. **Limbajul de programare ales**
3. **Fundamente teoretice**
4. **Proiectarea aplicatiei**
   1. **Descrierea arhitecturii sistemului**
   2. **Identificarea functionalitatilor aplicatiei**
   3. **Diagrama de clase**
   4. **Diagrame de interactiune**
      1. **Diagrama de activitate**
      2. **Diagrama de secventa**
      3. **Diagrama de colahorare**
   5. **Detalii de implementare**
   6. **Cazuri de testare**
5. **Testarea propriu-zisa a aplicatiei**
6. **Manual de instalare si utilizare**
7. **Concluzii si dezvoltari ulterioare**
8. **Bibliografie**
9. Specificatii si analiza sistemului

**Obiectivul principal** al temei este cel de a realiza o aplicație desktop pentru editarea de imagini. Această aplicație este capabilă să efectueze operații simple pe imagin. Aplicația este proiectată conform paradigmelor programării orientate pe obiecte folosind limbajul de programare Java. Pentru organizarea pe pachete și clase s-a folosit modelul arhitectural **Layers**. Punem în evidență mai multe obiective secundare, părți constituente ale obiectivului principal sus-menționat:

* **Divizarea responsabilităților pe clase și metode**, ca bună practică în proiectarea aplicației, prin structurarea pe pachete de clase conform modelului arhitectural **Layers**, pentru a facilita înțelegerea programului și dezvoltarea ulterioară – va fi detaliată în capitolul 4
* **Realizarea diagramelor UML** (diagrame de clase, de pachete, etc), pentru a facilita proiectarea ulterioară și transpunerea funcționalităților în cod
* **Lucrul cu fișiere și realizarea unui executabil .jar**
* **Lucrul cu diferite formate de imagini**
* **Implementarea operatiilor pe imagini**

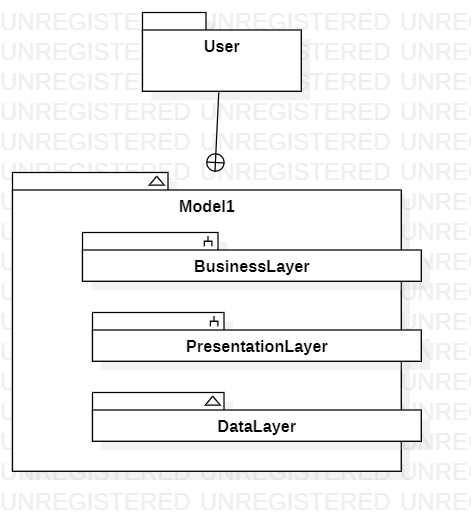
1. Limbajul de programare ales

In dezvoltarea acestei aplicatii am folosit limbajul de prorgramare Java, intrucat este un limbaj usor de folosit, multi-platforma si este foarte popular. De asemenea, exista o multitudine de librarii dezvoltate in limbajul Java pentru procesarea imaginilor. Sistemul de operare pentru care este proiectata aplicatia este Windows, un sistem de operare foarte popular in randul utilizatorilor de calculatoare personale.

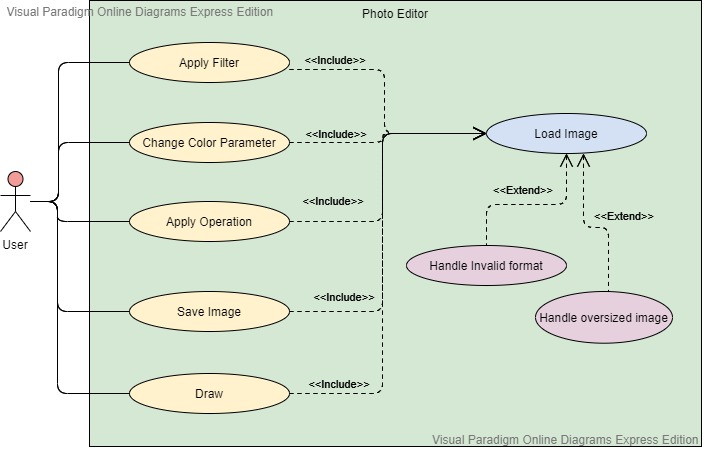
1. Fundamente teoretice

Pentru realizarea acestei aplicatii, a fost nevoie de studierea atat a algoritmilor de procesare de imagini, cat si a librariilor disponibile in limbajul Java pentru indeplinirea functiilor dorite. Atat librariile folosite, cat si resursele bibliografice folosite se regasesc in sectiunea de bibliografie.

1. Proiectarea aplicatiei
   1. **Descrierea arhitecturii sistemului**  
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
        
      Diagrama de pachete de mai sus înfățișează structura pachetelor importante din implementarea aplicației. Această aplicație este structurată conform modelului arhitectural Layers, ce presupune împărțirea responsabilităților pe mai multe nivele, lucru care facilitează înțelegerea și dezvoltarea ulterioară a aplicației.



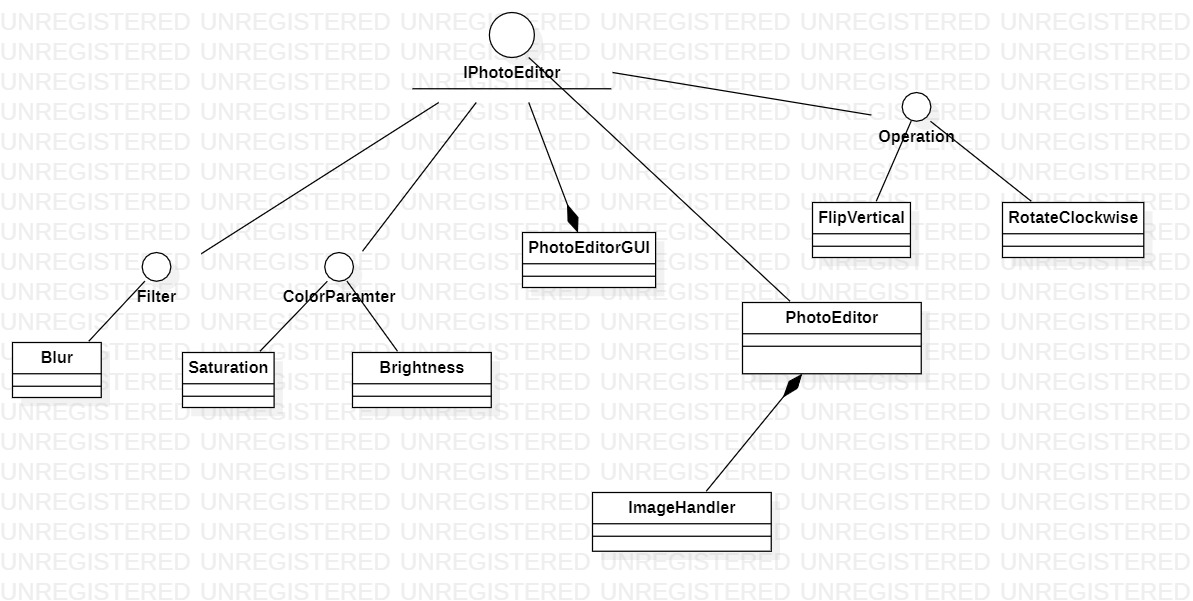
* 1. **Identificarea funcționalităților aplicației  
       
       
      Diagrama use-case a aplicației**



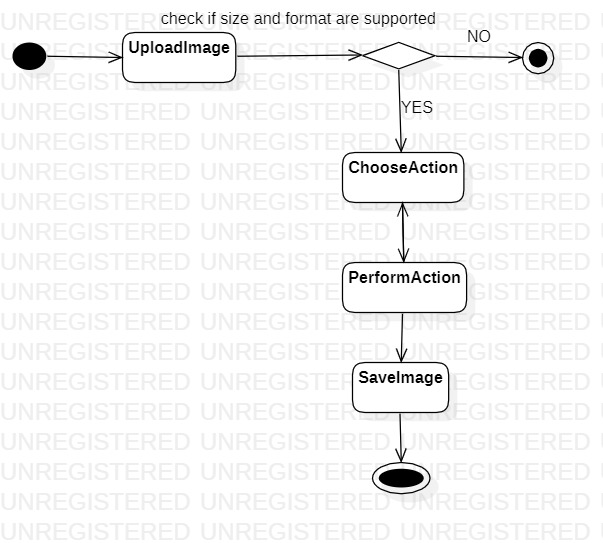
Se vor prezenta funcționalitățile oferite de aplicație, precum și detalierea cazurilor de utilizare a acesteia.

**Aplicația** prezintă următoarele funcționalități de editare de imagini:

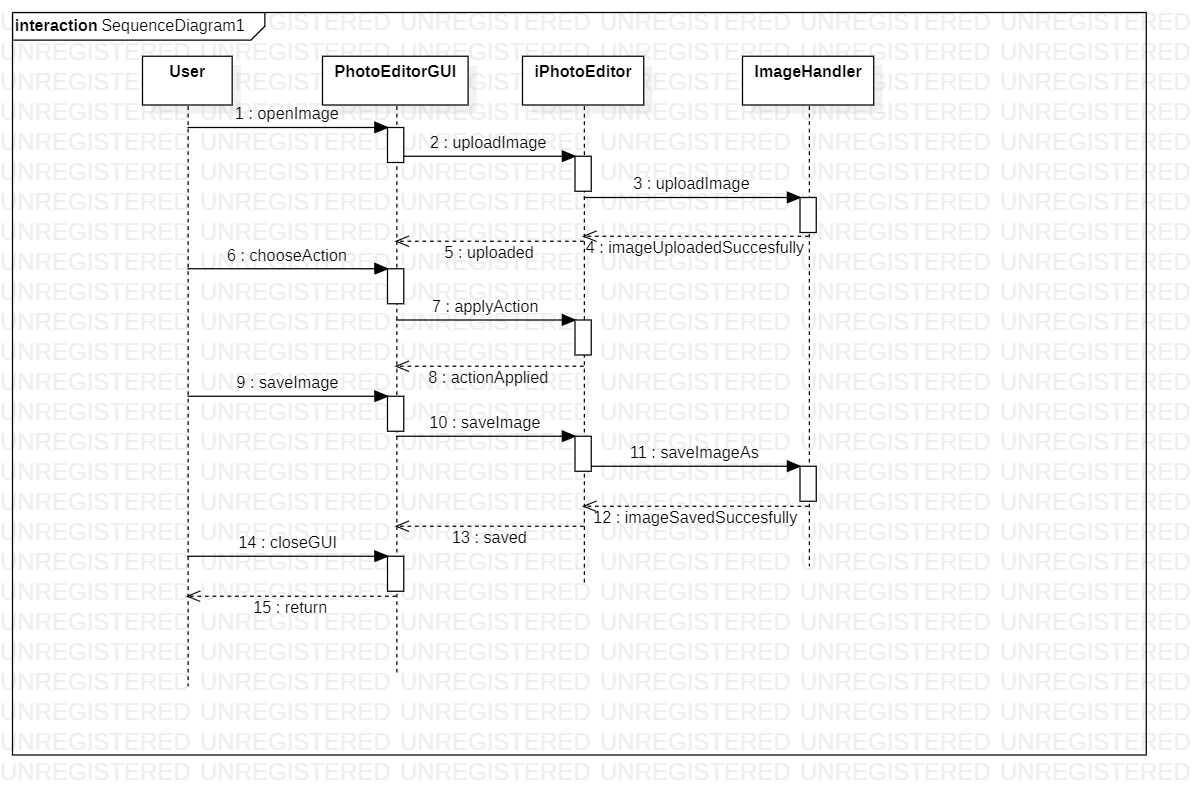
* Încărcarea de imagini în diferite formate: PNG, JPEG;
* Aplicarea de transformări simple pe imagini: decupare, rotație, reflexie;
* Aplicarea de filtre pe imagini;
* Modificarea parametrilor de culoare (luminozitate, contrast, saturație etc);
* Desenare pe imagini, cu diferite tipuri de pensule;
* Redimensionare imagini;
* Salvarea imaginilor pe disc, în diferite formate.
  1. **Diagrama de clase**



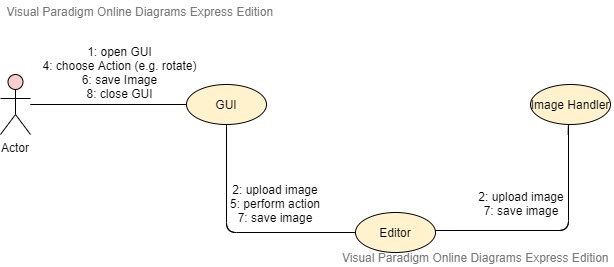
* 1. **Diagrame de interacțiune**
     1. **Diagrama de activitate**



* + 1. **Diagrame de secvență**



* + 1. **Diagrame de colaborare**



* 1. **Detalii de implementare**

Se vor prezenta în detaliu clasele definite în structura proiectului, evidențiind aspectele de proiectare OOP ale aplicației, precum și justificări legate de alegerile făcute în contextul proiectării. Fiecare pachet este descris împreună cu toate clasele și responsabilitățile acestora după cum urmează:

* + - 1. **Pachetul Presentation** – contine partea de interfata gafica dintre aplicatie si utilizator. Pentru proiectarea interfetei, s-a folosit JavaFX impreuna cu aplicatia SceneBuilder.
      2. **Pachetul Business –** contine logica de business a aplicatiei.
      3. **Pachetul Data –** implementeaza caile de comunicare cu sistemul de operare
         * **Clasa ImageHandler –** ofera metode prin care se pot incarca si salva imagini pe disc.
      4. **Pachetul Main**
         * **Clasa MainClass**
  1. Cazuri de testare

1. Testarea propriu-zisa a aplicatiei
2. Manual de instalare si utilizare
3. Concluzii si dezvoltari ulterioare
4. Bibliografie

[1] <https://www.w3schools.com/java/java_intro.asp>

[2] <https://en.wikipedia.org/wiki/Singleton_pattern>

[3] <https://www.baeldung.com/java-pdf-creation>

[4] <https://www.jetbrains.com/help/>

[5] <https://www.udemy.com/>

[6] <https://www.geeksforgeeks.org/>

[7] <https://www.w3schools.com/>

[8] <https://stackoverflow.com/>

. . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . .