



Documentatie Tehnica

(Technical Documentation)

Version 1.0

14 January, 2013

Echipa Blue, Document Processing Manager

Cuprins

Cuprins.....	2
1. Scopul documentului	3
2. Continutul documentului	3
3. Descrierea generala a produsului	4
3.1 Descrierea temei.....	4
3.2 Misiunea proiectului	4
4. Modelul architectural.....	5
4.1 Arhitectura sistemului	5
4.2 Tehnologiile folosite	6
5. Cerinte functionale si nefunctionale	7
5.1 Cerinte functionale	7
5.2 Cerinte nefunctionale.....	11
6. Modelul interfetei cu utilizatorul	11

1. Scopul documentului

Acest document are ca scop descrierea intr-un mod realist si cat mai complet a interfetei necesare procesarii de imagini corespunzatoare proiectului Preprocessing GUI.

In acest document sunt prezentate comportarea externa a aplicatiei, cerintele functionale si nefunctionale, restrictiile de proiectare si alti factori necesari pentru descrierea completă a cerințelor software

Documentul serveste drept ghid in solutia abordata de catre echipa de dezvoltare.

2. Continutul documentului

Documentul este impartit in VI sectiuni dupa cum urmeaza:

- I sectiune prezinta obiectivele propuse de echipa
- A II-a sectiune descrie pe scurt principalele parti ale prezentei documentatii
- A III-a sectiune prezinta descrierea generala a produsului
- A IV-a sectiune prezinta modelul arhitectural
- A V-a prezinta cerintele functionale si nefunctionale
- A VI-a sectiune prezinta workflowul aplicatiei prin prisma unor capturi de imagini a aplicatiei.

3. Descrierea generala a produsului

3.1 Descrierea temei

Echipa Managementul Proiectelor Software din cadrul Facultatii de Automatica si Calculatoare si-a propus dezvoltarea unui proiect care sa ofere o solutie pentru managementul unui intreg workflow al procesarilor de documente din format fizic, printat, in format digital. Avand in vedere dimensiunea aplicatiei si avand la dispozitie o parte din executabile, solutia propusa pentru realizarea acestui obiectiv reprezinta un sistem OCR.

Un sistem OCR este o aplicatie complexa, ce presupune interconectarea a numeroase componente ce oferă facilitati variate. In mod ideal, odata realizate setarile pentru rulara aplicatiei, imaginile document ar trebui sa ruleze „batch mode”, fara nicio intervenție externă. Acest lucru este posibil, dar rezultatele, nu sunt perfecte. Asadar este adesea nevoie de a face unele corectii in ceea ce priveste rezultatele obtinute.

Proiectul a fost impartit in parti echivalente din punctul de vedere al efortului de dezvoltare si distribuit echipelor.

Documentul curent se ocupa de prezentarea celei de-a III-a parti a proiectului principal, T3-Preprocessing GUI.

3.2 Misiunea proiectului

Impreuna cu echipa, am stabilit de comun acord principalele obiective de atins ale aplicatiei pe care am dezvoltat-o:

- stabilirea parametrilor necesari preprocesarii prin parsarea unor documente xsd
- evaluarea rezultatele diferitelor preprocesari asupra imaginilor;
- crearea unui xml valid si corect corespunzator xsd-ului pus la dispozitie si care sa corespunda parametrilor finali ai imaginilor, rezultati in urma preprocesarii.

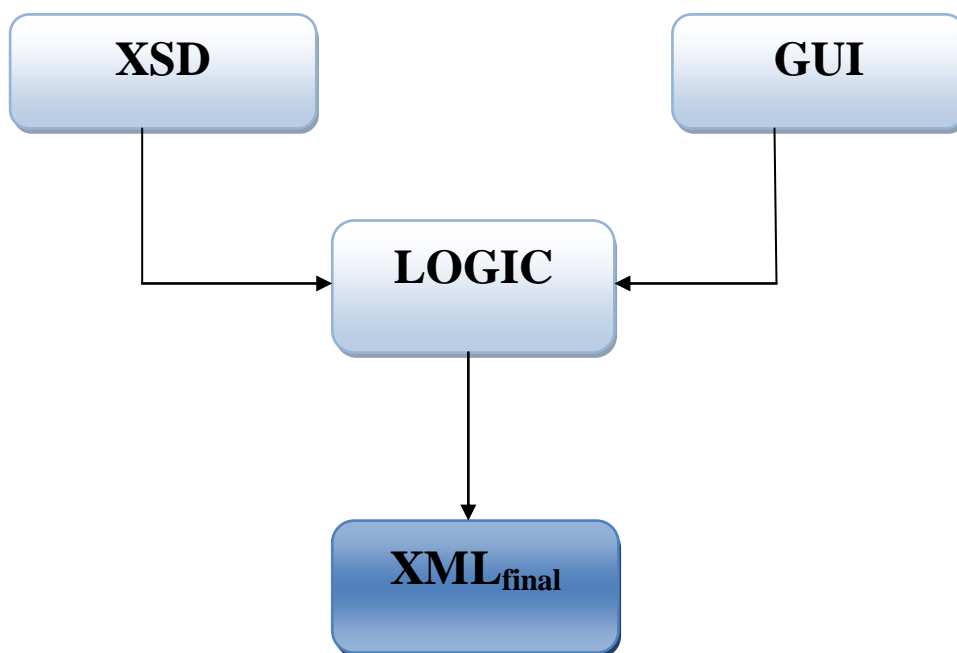
4. Model arhitectural

4.1 Arhitectura sistemului

Solutia software a proiectului se bazeaza pe sablonul de event-driven architecture.

Acest sablon promoveaza producerea, detectarea si reactia la evenimente. In cazul aplicatiei noastre evenimentele vor fi generate prin selectarea unuia din filtrele de procesare disponibile aplicatiei si rulara acestuia.

Diagrama de arhitectura, prezentata in continuare, ilustreaza componentele aplicatiei si modul in care acestea interactioneaza pentru obtinerea rezultatului dorit.



Aplicatia prezinta urmatoarele module:

- **Modulul GUI (Interfata grafica)** se ocupa de desenarea si afisarea ferestrei principale cat si a ferestrelor intermediare de preluare a datelor de intrare de la utilizator.
- **Modulul de Logica (Business Logic)** se ocupa de coordonarea celorlalte module astfel incat rezultatele rularii lor sa conduca la obtinerea unui fisier xml conform cu specificatiile din fisierul xsd.
- **Modul Parser** se ocupa de parsarea fisierelor xsd cu scopul de a stabili parametrii necesari unui anumit filtru si de generarea xml-ului final .

Interactiunea dintre componente

In momentul pornirii aplicatiei de catre utilizator este apelata clasa Main din Modulul de Logica. Aceasta va afisa fereastra principala a aplicatiei, corespunzatoare modulului de GUI.

In acest moment utilizatorul introduce in interfata imaginea pe care doreste sa o proceseze si poate selecta filtrele pe care i le va aplica. Este apelat in continuare Modulul Parser in care se determina numele imaginii si numele si tipul parametrilor necesari filtrului dorit. Valorile acestora vor fi introduse de catre utilizator in interfata grafica. Dupa rularea aplicatiei se va obtine un fisier xml care trebuie sa corespunda standardelor si restrictiilor precizate in fisierul xsd, precum si o imagine preprocesata ce poate fi salvata in calculator si asupra careia putem aplica alte filtre.

4.2 Tehnologiile folosite

Proiectul a fost dezvoltat in intregime in Eclipse, in limbajul Java.

In continuare voi prezenta o scurta descriere a tehnologiilor complementare utilizate in procesul de dezvoltare al proiectului:

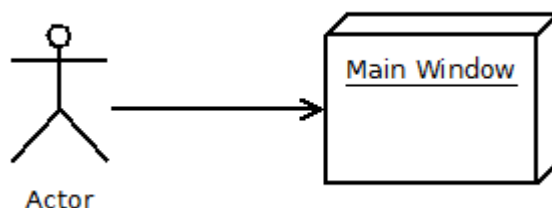
- ❖ Pentru interfata grafica:
 - *SWT*- este un toolkit ce se integreaza in platform java si a fost folosit pentru dezvoltarea ferestrelor principale ale aplicatiei
 - *SWING* – folosit pentru realizarea ferestrelor de parametric necesare introducerii valorilor in vederea procesarii imaginii
- ❖ Pentru partea de parsare xsd:

- *Bibliotecile XSOM si relaxngDatatype* – ele au fost folosite pentru crearea unui parser care sa scoata parametrii necesari unui anumit tip de procesare
- ❖ Pentru partea de comparare a 2 imagini:
 - *ImageJ* – biblioteca folosita pentru compararea a 2 imagini, ofera posibilitati de analiza a si procesare a imaginii, de calculare de suprafete, de distante, unghiuri s.a.

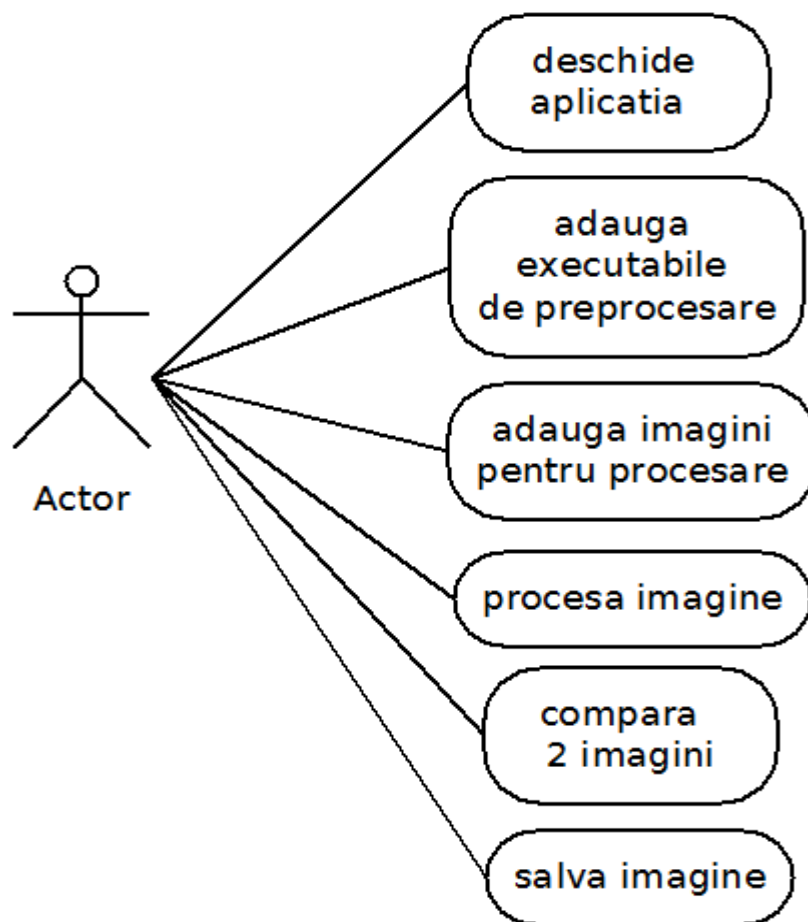
5. Cerinte functionale si nefunctionale

5.1 Cerinte functionale

Workflowul aplicatiei prezinta drept actori, pe utilizatorii care, prin intermediul aplicatiei si a executabilelor de procesare, aplica diverse filtre si procesari asupra imaginilor.



Use Case Utilizator:



Use Case pentru “deschide aplicatia”

Actor	Utilizator
Scop	Procesarea unor imagini
Preconditii	Existenta aplicatiei
Scenariu	Ruleaza clasa Main din pachetul Main
Exceptii	Aplicatia crapa

Use Case pentru “adauga executabile de preprocesare”

Actor	Utilizator
Scop	Adaugarea unor executabile de procesare a imaginilor
Preconditii	Existenta aplicatiei, aplicatia este deschisa, sa existe executabilele default in workspace
Scenariu	Din meniul File, din fereastra principala se selecteaza Add Default Executables sau Import executables
Exceptii	Nu exista executabilele default sau nu au atat xsd cat si exe pentru un executabil

Use Case pentru “adauga imagini pentru preprocesare”

Actor	Utilizator
Scop	Adaugarea unei imagini
Preconditii	Existenta aplicatiei, aplicatia este deschisa, sa existe imagini in workstation-ul utilizatorului
Scenariu	Din meniul File, din fereastra principala se selecteaza Load image, sau direct din fereastra principala se apasa pe butonul Load image din panelul stang
Exceptii	Nu exista imagini in calculator

Use Case pentru “proceseaza imagini”

Actor	Utilizator
Scop	Procesarea imaginii adaugate
Preconditii	Existenta aplicatiei, aplicatia este deschisa, sa existe imagini in workstation-ul utilizatorului, sa fie adaugate executabilele de procesare
Scenariu	Se selecteaza din caseta de executabile unul sau mai multe executabile si se apasa butonul Apply. Apoi sunt adaugate valori in casetele de parametrii aferente executabilelor selectate si se apasa Process.
Exceptii	Nu exista imagini incarcate sau nu exista executabilele.

Use Case pentru “compararea a doua imagini”

Actor	Utilizator
Scop	Compararea a 2 imagini
Preconditii	Existenta aplicatiei, aplicatia este deschisa, sa existe imagini in workstation-ul utilizatorului
Scenariu	Se apasa butonul Compare din fereastra principala, se incarca 2 imagini ce se doresc a fi comparate in noua fereastra si se apasa butonul Compare din aceasta ultima fereastra.
Exceptii	Nu exista imagini incarcate .

Use Case pentru “salvare imagine”

Actor	Utilizator
Scop	Salvarea imaginii preprocesate
Preconditii	Existenta aplicatiei, aplicatia este deschisa, sa fi procesat imaginea

Scenariu	Se apasa butonul Save image din panelul drept al ferestrei principale sau optiunea Save image din meniul File.
Exceptii	Nu exista imagini preprocesate.

5.2 Cerinte nefunctionale

Proiectul este functional atat pe Windows cat si pe Linux.

Pentru a putea rula trebuie sa existe instalate:

-un ide (ex. Eclipse)

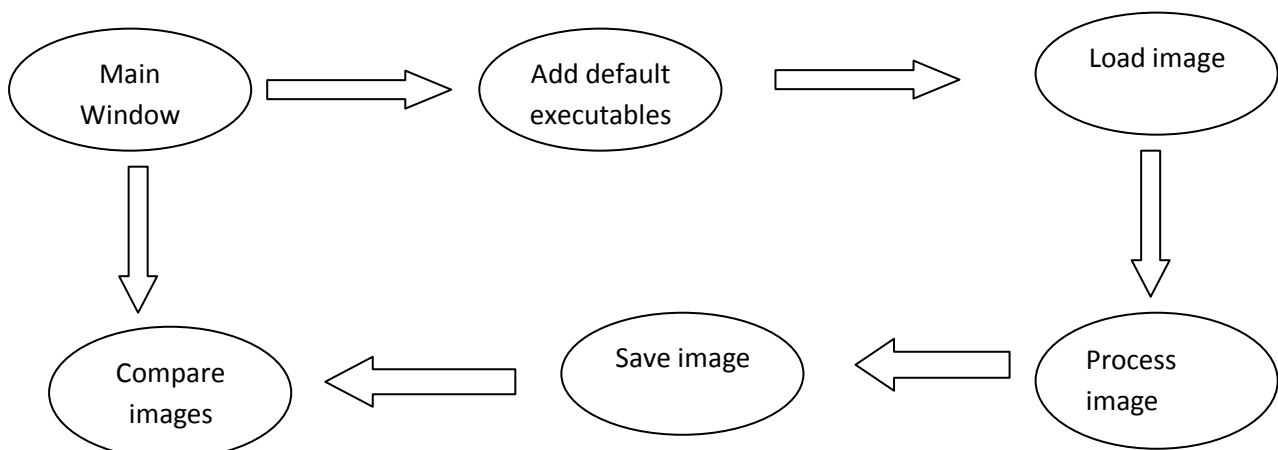
- JDK 1.6

In ceea ce priveste cerintele hardware, este necesar ca workstationul sa aiba:

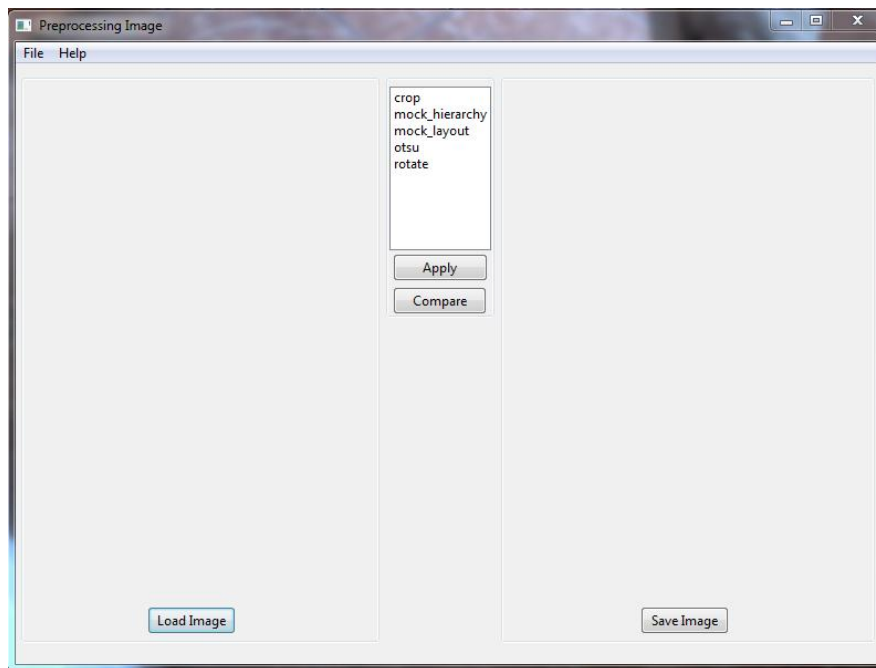
- CPU 1 Ghz
- RAM 1GB
- HDD Space 512MB

6. Modelul interfetei cu utilizatorul

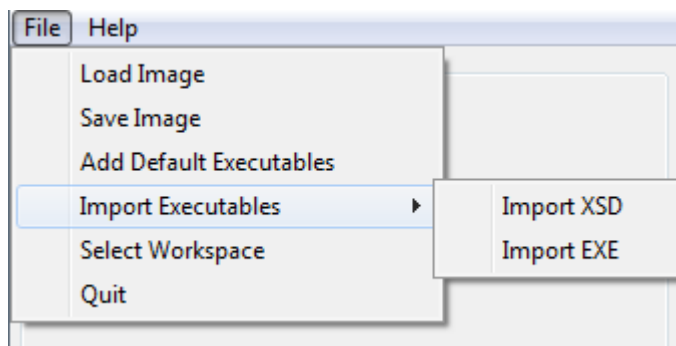
In cadrul modului GUI, ferestrele aplicatiei se succed respectand workflowul descris in proiect.



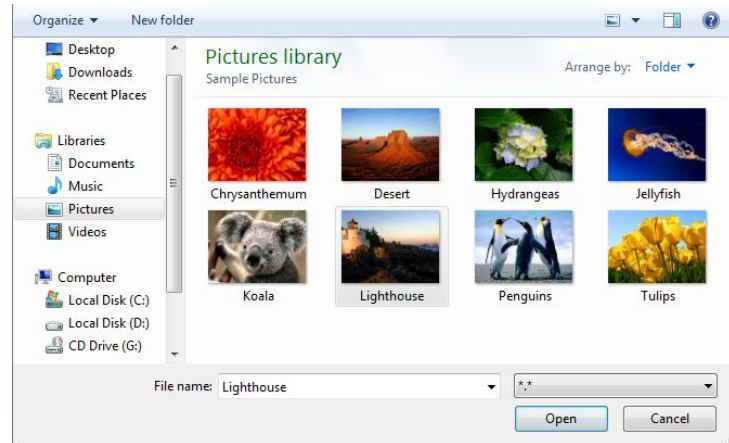
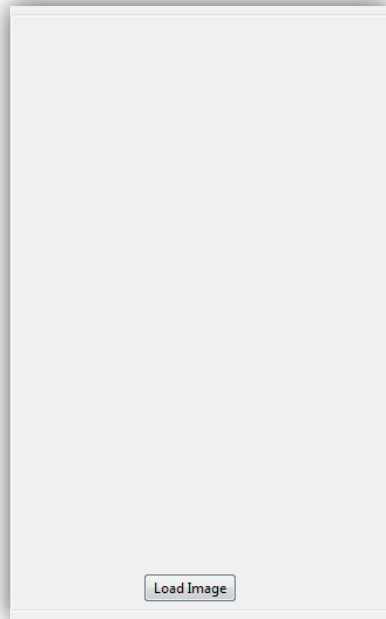
Fereastra principala



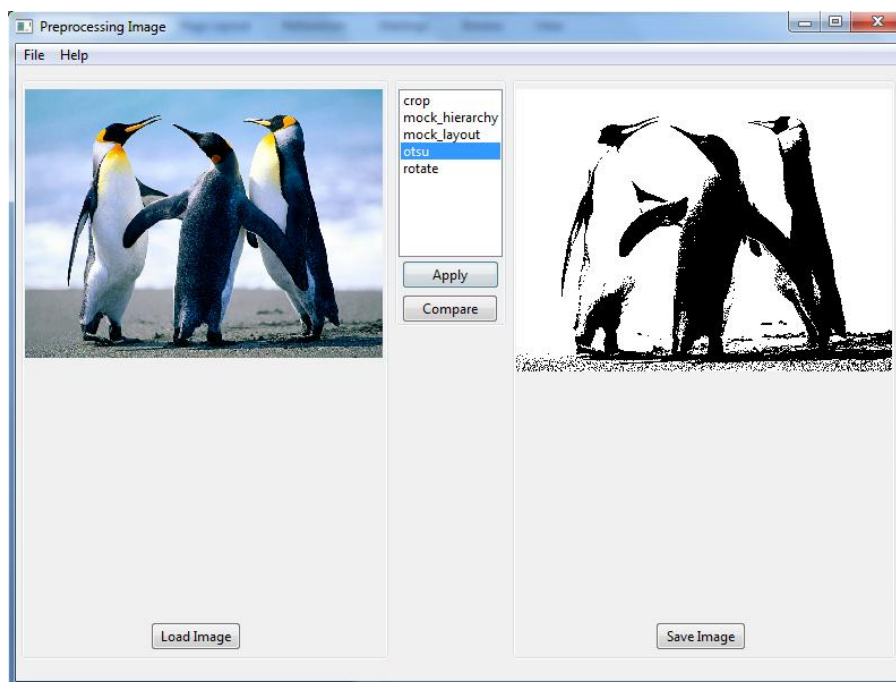
Add default Executables



Load image



Process Image & Save Image



Compare images

