

# Document de Proiectare a Soluției Aplicației Software

(Software Design Document)

## Layout Analysis

Echipa "Echipa T4 de miercuri de la 8 la 10"

Facultatea de Automatică și Calculatoare, Universitatea Politehnică București

Versiunea 1.0

23 Octombrie, 2012

## Cuprins

Cuprins	2
1. Scopul documentului	3
2. Continutul documentului	3
3. Modelul datelor	3
3.1. Structuri de date globale	3
3.2. Structuri de date de legatura	4
3.3. Structuri de date temporare	4
3.4. Formatul fisierelor utilizate	4
4. Modelul arhitectural si modelul componentelor	5
4.1. Arhitectura sistemului	5
4.1.1. Sabloane arhitecturale folosite	5
4.1.2.1. Diagrama de arhitectura	6
4.1.2.2. Detalierea arhitecturii componentelor	7
4.2. Descrierea componentelor	8
4.3. Restrictiile de implementare	8
4.4. Interactiunea dintre componente	9
5. Modelul interfetei cu utilizatorul	9
5.1. Succesiunea interfetelor	9
5.2. Ferestrele aplicatiei	11
5.2.1. Fereastra de selectare a documentului	11
5.2.2. Editarea caracterelor în interiorul liniilor	12
5.2.3. Editarea liniilor în paragrafe	12
5.2.4. Editarea paragrafelor în document	14
5.2.5. Ecranul de paginare și generare	15
6. Elemente de testare	16
6.1. Componente critice	16

## 1. Scopul documentului

Acest document are ca scop descrierea și arhitecturarea soluției software intitulată "Layout Analysis" cu scopul de a corecta erorile din gruparea caracterelor în rânduri și a rândurilor în blocuri, paginării.

Documentul descrie clar, complet și concis obiectivele proiectului, stând la baza contractului de execuție a proiectului și servind drept ghid unic de proiectare și implementare a soluției pentru echipa de dezvoltatori.

## 2. Conținutul documentului

Secțiunile principale ale documentului sunt cele ce urmează:

- Modelul datelor - reprezintă principalele structuri de date folosite.
- Modelul arhitectural și al componentelor - reprezintă arhitectura sistemului, descrie modulele componente și afișează diferite șabloane arhitecturale și/sau de programare.
- Modelul interfeței cu utilizatorul - prezintă modul prin care un utilizator interacționează cu aplicația precum și succesiunea ferestrelor în fluxul de aplicație.
- Modelul de testare - prezintă componentele cu potențial mare de eșec și alternative de proiectare a acestora.

## 3. Modelul datelor

### 3.1. Structuri de date globale

Structura globală de date este un arbore, reprezentând o abstractizare optimă a organizării și a conținutului fișierului .xml de ieșire, conform cu schema aferentă. Astfel, acesta va fi format din mai multe tipuri de noduri:

- *DocumentNode* - nodul corespunzător tag-ului <Document> (rădăcina .xml-ului)
  - *ComposedBlockNode* - noduri corespunzătoare tag-urilor <ComposedBlock> ;
- fiecare nod are ca părinte rădăcina

- *TextBlockNode* - noduri corespunzătoare tag-urilor <TextBlock> ; fiecare nod are ca părinte fie rădăcina, fie un nod de tip *ComposedBlockNode*
- *ImageBlockNode* - noduri corespunzătoare tag-urilor <ImageBlock> ; fiecare nod are ca părinte rădăcina
- *TextLineNode* - noduri corespunzătoare tag-urilor <TextLine> ; fiecare nod are ca părinte un nod de tip *TextBlockNode*
- *StringNode* - noduri corespunzătoare tag-urilor <String> ; fiecare nod are ca părinte un nod de tip *TextLineNode*.

Rădăcina arborelui este unică și este întotdeauna un nod de tip *DocumentNode*.

Frunzele arborelui (nodurile terminale) sunt fie noduri de tip *ImageBlockNode*, fie *StringNode* și conțin datele corespunzătoare unei imagini, așa cum au fost specificate în fișierul .xml de intrare, respectiv textul parsat și eventual corectat de către utilizator.

### 3.2. Structuri de date de legătură

Legătura dintre reprezentarea internă, arborescentă, a datelor și cea vizuală, care ține de modulul GUI, se va face prin intermediul câmpurilor de text afișate pe ecran (editabile sau needitabile de către utilizator) și a formelor geometrice (chenare / highlight-ing) ce delimitează zonele de text. Din modulul GUI se vor transmite modulului de logică date precum coordonatele selecției cu mouse-ul, indexul blocului selectat, indexul unei linii marcată pentru eliminare etc. Modulul de logică transmite modulului de procesare XML ierarhia datelor, precum și datele propriu-zise. De asemenea, modulul de procesare XML va comunica și cu modulul GUI în situația în care utilizatorul încarcă pentru editare un fișier .xml; în acest caz, se vor transmite atât datele efective - textul - cât și coordonatele acestuia în pagină și organizarea pe blocuri.

### 3.3. Structuri de date temporare

Structurile de date temporare utilizate vor avea rol strict auxiliar și nu vor consuma resurse semnificative de memorie și/sau efort computațional.

### 3.4. Formatul fișierelor utilizate

Fișierele de intrare pot fi:

- documente .xml valide și bine formate, conforme cu schema .xsd prestabilită;

- imagini în format .tiff îndreptate în așa fel încât textul să fie orizontal și cu contrastul ajustat în așa fel încât textul să fie ușor lizibil.

Fișierele de ieșire sunt documente .xml valide și bine formate, conforme cu schema .xsd prestabilită.

## 4. Modelul arhitectural și modelul componentelor

### 4.1. Arhitectura sistemului

#### 4.1.1. Șabloane arhitecturale folosite

Componenta de "Layout Analysis" a proiectului global respectă o arhitectură de tip "*batch processing*".

Datele primite la intrare trec printr-o serie de etape disjuncte până să ajungă în starea de date valide de ieșire.

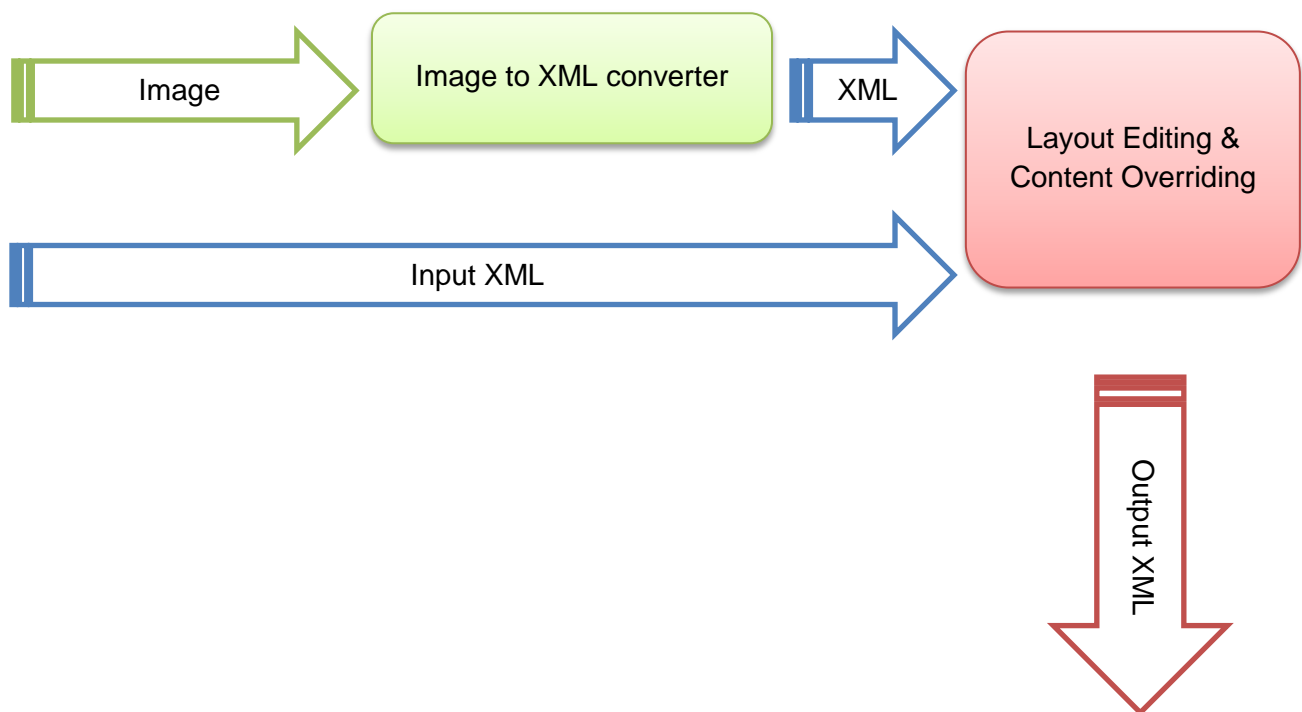
Fiecare etapă a procesării are atribuit un modul aplicație care respectă un protocol aplicație concretizat printr-o serie de interfețe de programare și date, încât imlementarea să poată varia independent de interfață (conform principiului de "*black box*").

O astfel de arhitectură oferă următoarele avantaje:

- cuplare redusă între componentele aplicației
- ușurință în implementarea unor noi elemente componente și intercalării lor datorită programării orientate pe interfețe
- extindere ușoară către arhitecturi de tip cluster ori multiprocesor datorită dependențelor de date reduse impuse de arhitectură
- arhitectură ușor inteligibilă, simplă, deci cu o curbă de învățare ne semnificativă

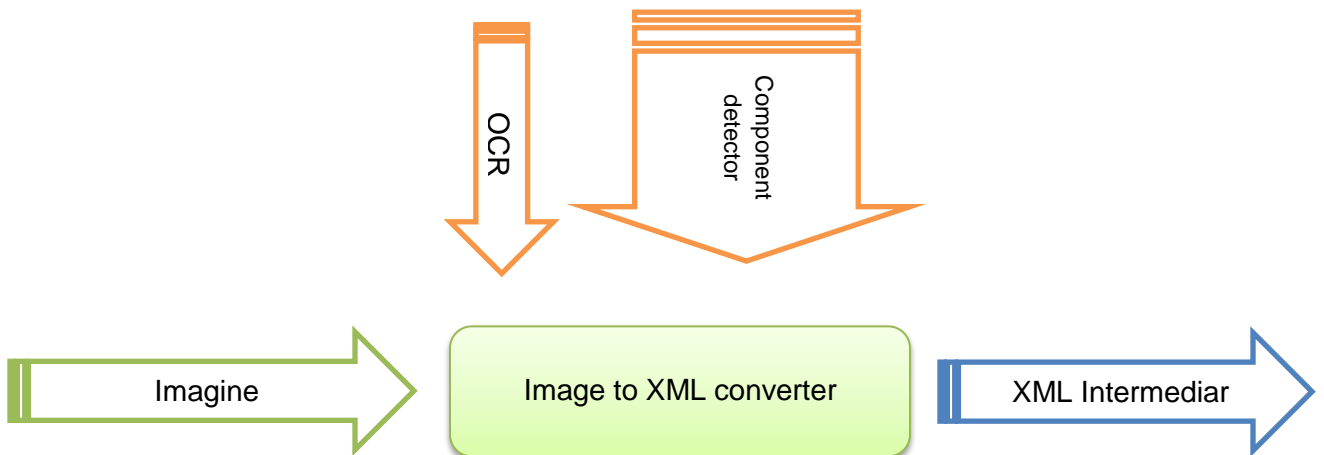
#### 4.1.2.1. Diagrama arhitecturală

Diagrama de arhitectură de mai jos descrie componentele arhitecturii aplicației și relațiile de interacțiune dintre acestea.

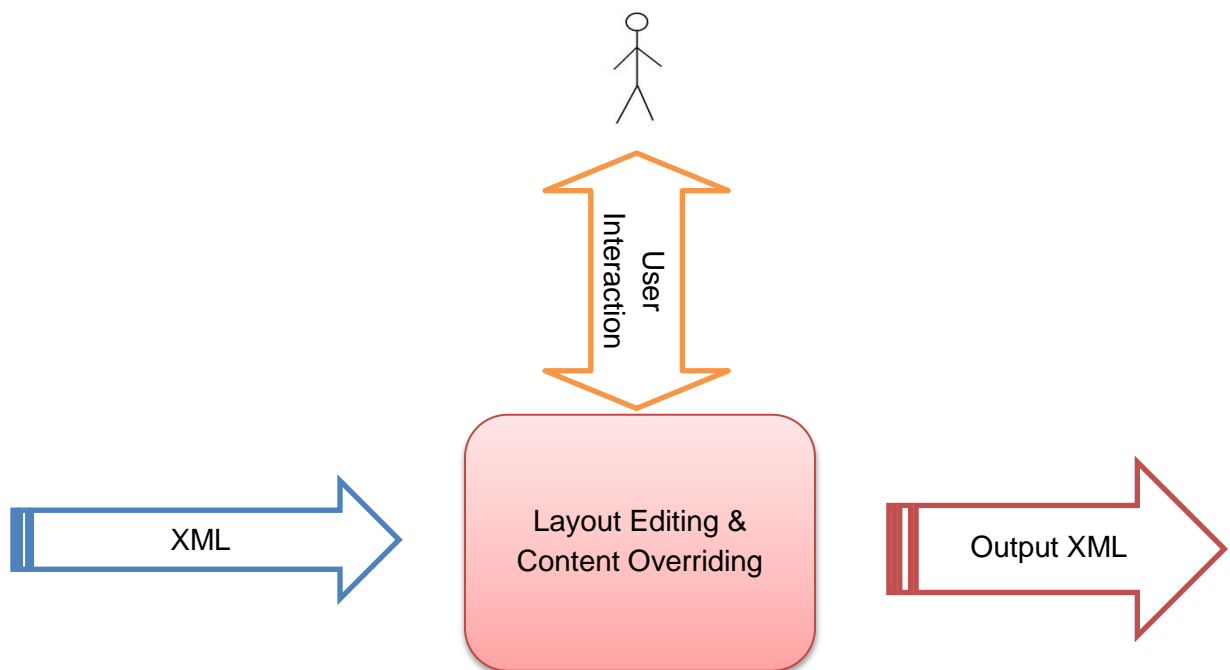


#### 4.1.2.2. Detalierea arhitecturii componentelor

Componenta de "Layout Synthesis".



Componenta de "Layout and Content Editing"



## 4.2. Descrierea componentelor

Aplicația cuprinde următoarele componente:

- *"Image to XML Converter"* (Convertor imagine - XML)  
Componenta primește input o imagine și are rolul de a efectua o analiză structurală în vederea generării unui fișer XML conform specificației XSD. Există două funcții majore implementate la nivelul aceste componente:
  - Recunoașterea caracterelor existente in imagine (OCR)
  - Gruparea acestora in structuri de layout conform specificației, împreună cu determinarea atributelor acestora (spațiu ocupat, orientare, componente copil)
- *"Layout Editing & Content Overriding"* (Editor grafic al aspectului si suprascriere de continut)  
Componenta realizează corecția unui document convertit la XML printr-o succesiune de interacțiuni cu utilizatorul. Interacțiunile pot avea următoarea natură:
  - Redimensionări și/sau divizări/uniri de elemente
  - Corecție text

## 4.3. Restricțiile de implementare

Modulele aplicației trebuie să acopere următoarele restricții de implementare:

- Portabilitate peste sistemele de operare majore (Windows, Linux, Mac)
- Modulul trebuie să poata fi operat (executat) în consistență cu specificația, respectând interfețele asociate
- Datele de intrare și ieșire trebuie să fie conforme cu specificația XSD
- Toate modulele aplicației vor fi dezvoltate utilizând limbajul de programare Java Standard Edition Versiunea 7.



## 4.4. Interacțiunea dintre componente

Rularea aplicației va aduce utilizatorul în fața unei alegeri privind tipul datelor de intrare:

- Imagine
- XML

În cazul selectării tipului *"image"*, se va apela modulul de conversie la XML. În urma operației se va obține un fișier XML conform specificației XSD, care va centraliza rezultatele detecției de componente și detecției de text.

Dupa acest pas, sau în cazul în care datele sunt prezentate deja sub în format XML, se lanseaza modulul de editare si corecție.

Interfața GUI pune la dispoziția utilizatorului un set de operații conform cu specificația pentru manipularea componentelor documentului, acestea includ:

- Redimensionări
- Unificări
- Diviziuni
- Suprascrisere text
- Ștergere entități
- Adaugare entități

După această secvență de operații succesive se va exporta rezultatul către un fișier XML conform specificației XSD, constituind rezultatul transformărilor asupra intrării.

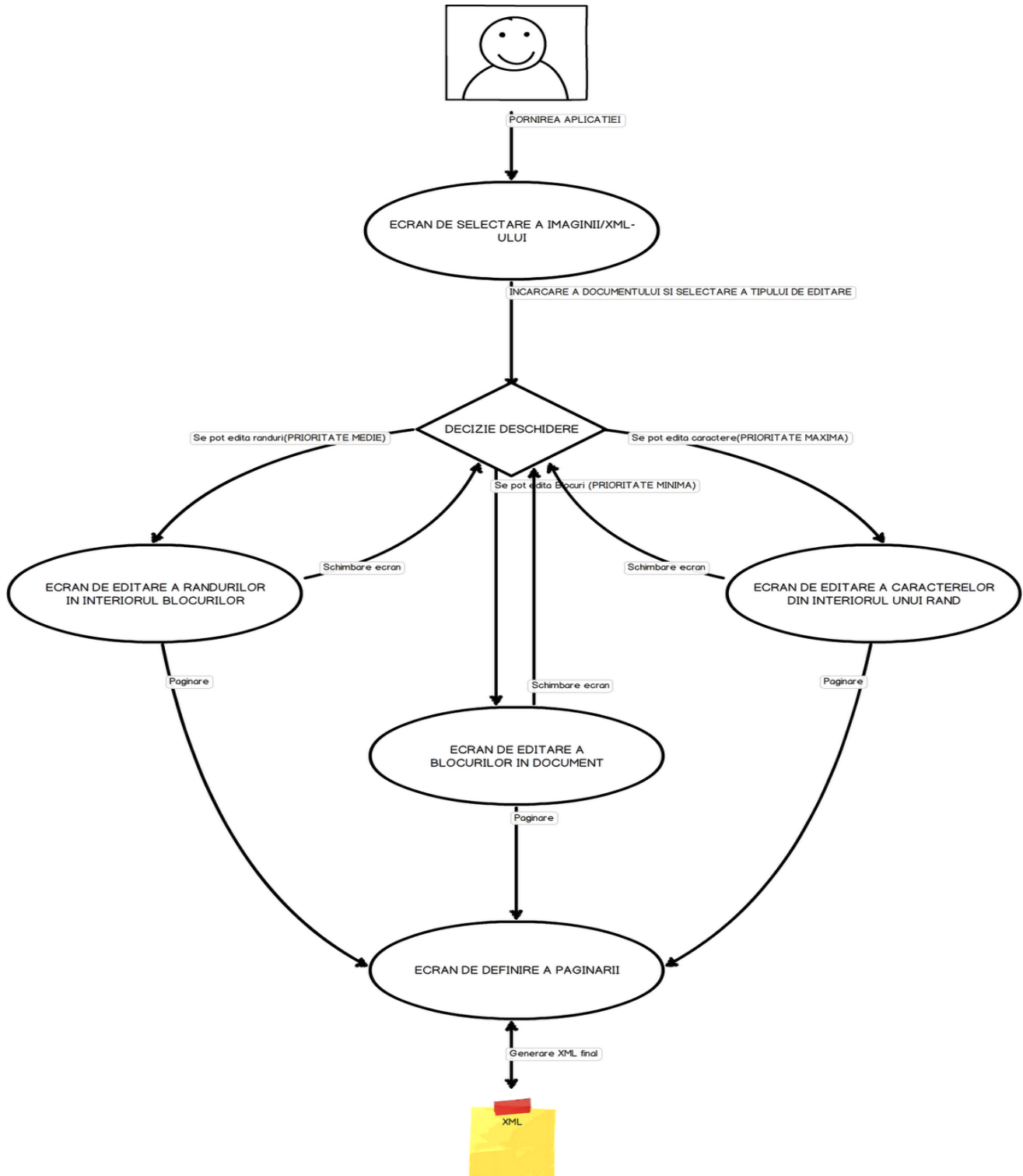
Utilizatorului i se va oferi optiunea de a începe o nouă secvență de operații sau de a părăsii aplicația.

## 5. Modelul interfeței cu utilizatorul

### 5.1 Succesiunea interfețelor

Modului GUI este reprezentat de o interfață Awt/Swing, respectând workflow-ul de editare și corectare a documentelor XML și de tip imagine stabilit în prezentul proiect.

Succesiunea ecranelor este descrisă de următoarea diagramă:



## 5.2 Ferestrele Aplicație

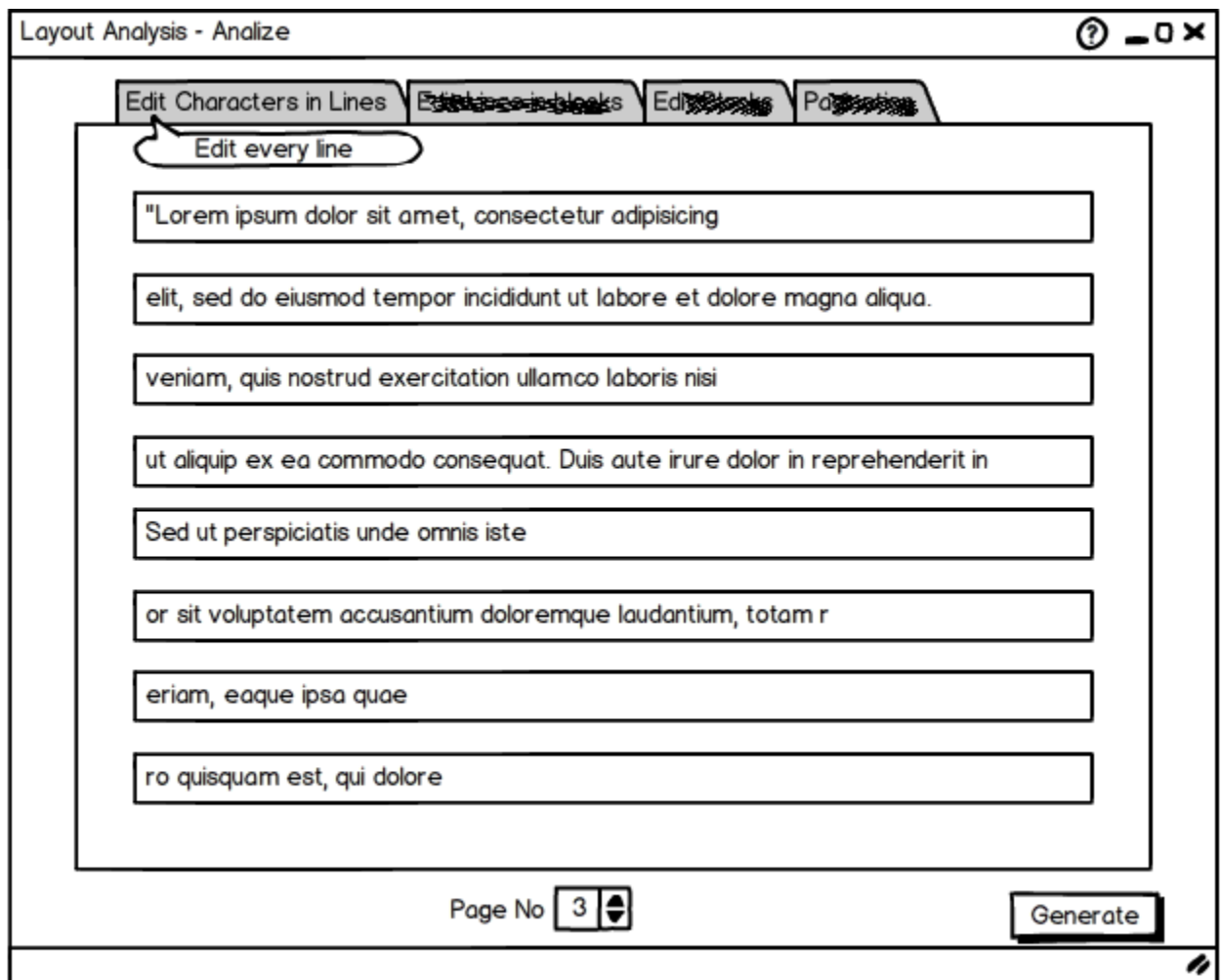
### 5.2.1 Fereastra de selectare a documentului

Această fereastră reprezintă dashboard-ul aplicației. Utilizatorul selectează fișierul pe care dorește să îl editeze / corecteze, alege tipul de editare pe care dorește să îl efectueze și apasă butonul "Analyze" pentru a începe editarea.

The screenshot shows a window titled "Layout Analysis - Selecting". The main heading inside the window is "Select your input file (XLS/IMG)". Below this heading, there is a text input field labeled "Path to input" and a "Browse" button. Underneath these, there is a section titled "Select what to edit" containing three checkboxes: "Characters" (checked), "Lines of Characters" (checked), and "Blocks of lines" (unchecked). At the bottom of the window is a large, rounded button labeled "Analyze".

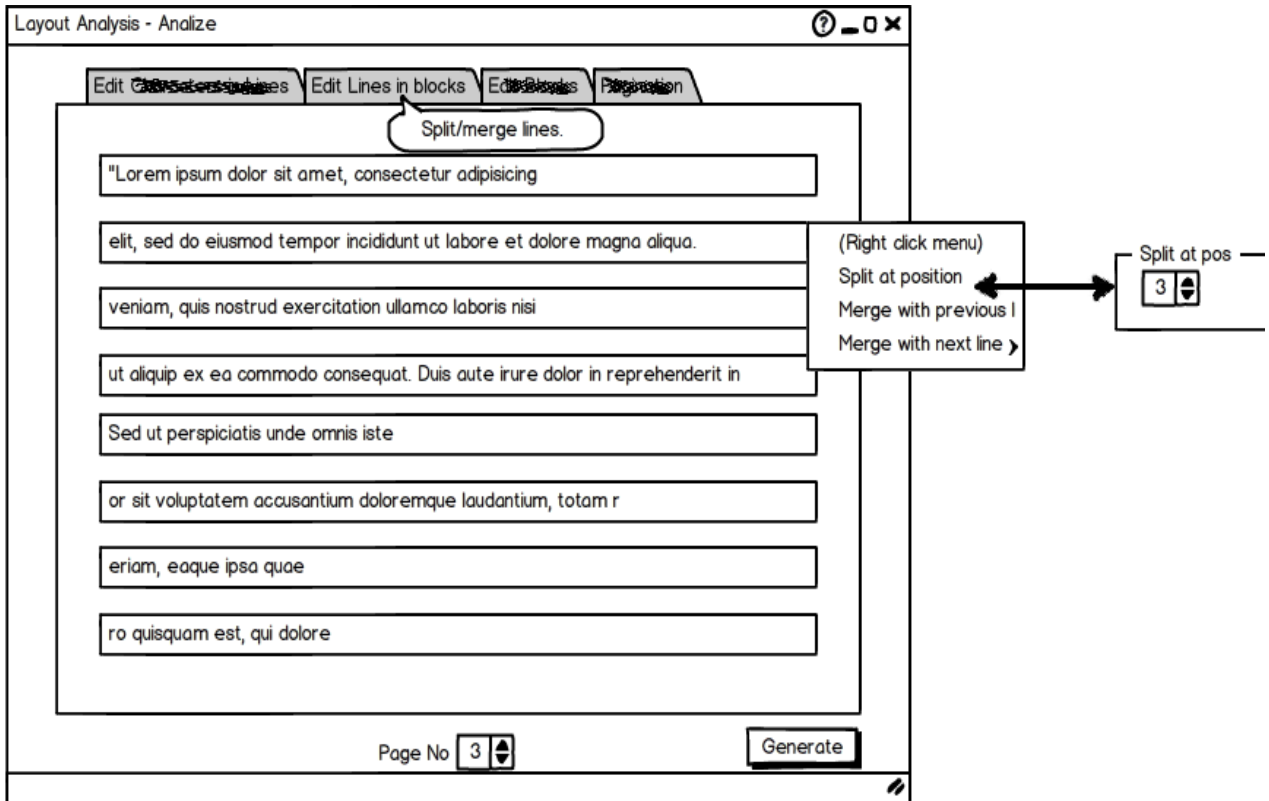
### 5.2.2 Editarea caracterelor în interiorul liniilor

În funcție de tipul de analiză pe care îl selectează din primul ecran, utilizatorul poate ajunge în ecranul de editare al caracterelor.



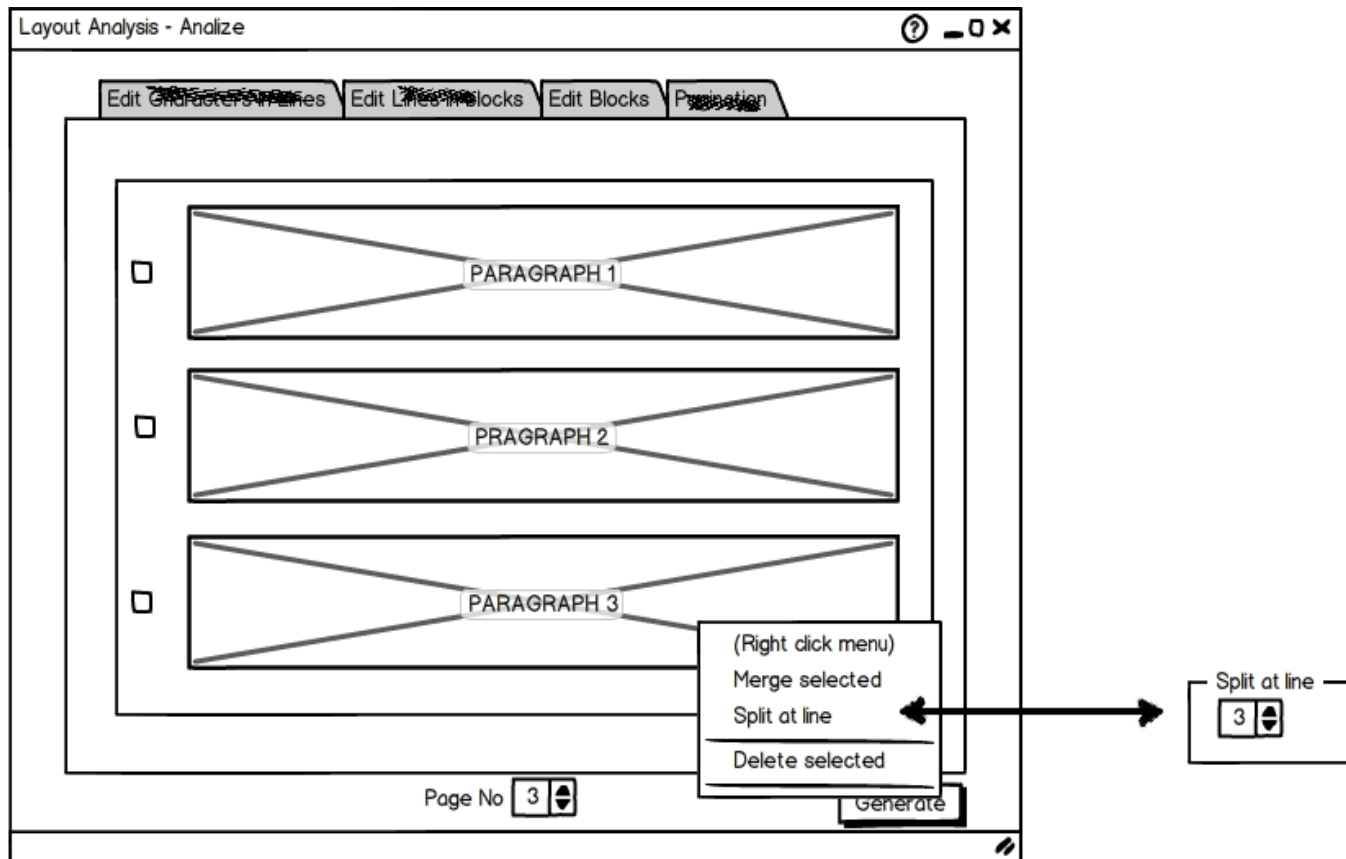
### 5.2.3 Editarea liniilor în paragrafe

Aici utilizatorul va putea să unească două linii (dacă sunt în același bloc) sau să împartă o linie în două.



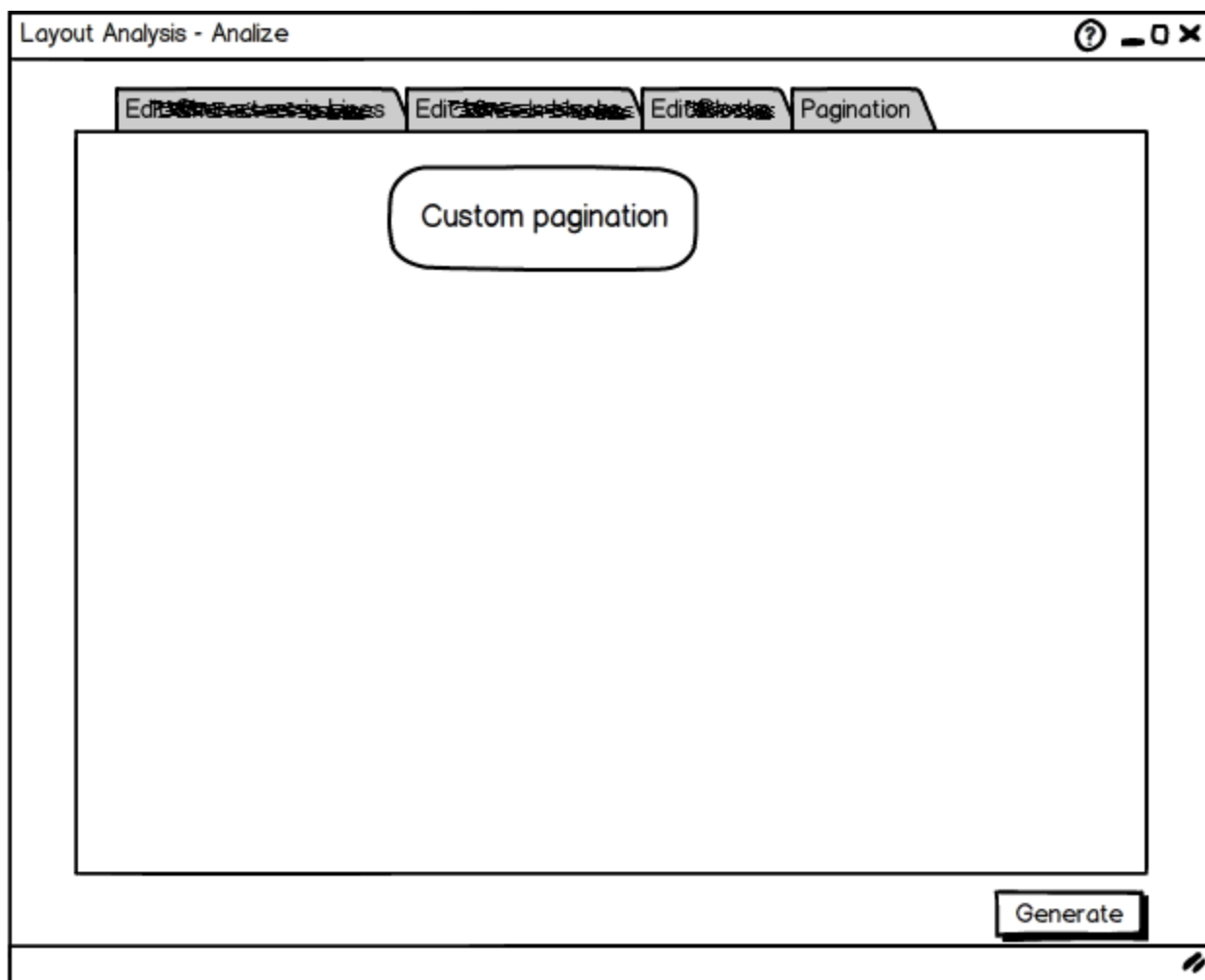
## 5.2.4 Editarea paragrafelor în document

Utilizatorul va putea să aleagă dintre a concatena sau a împărți paragrafele.



### 5.2.5 Ecranul de paginare și generare

Din acest ecran utilizatorul poate să suprascrie paginarea detectată de OCR și poate genera fișierul XML final.



## 6. Elemente de testare

Testarea se va face în mai multe etape, în funcție de evoluția proiectului:

1. Testarea elementelor de bază:

- verificarea interfeței grafice;
- parsarea corectă a XML-ului.

2. Testarea funcționalității:

- verificare output;
- timpul de procesare trebuie să fie unul mediu (propus de dezvoltatori).

3. Testare finală:

În acest moment proiectul trebuie să fie aproape de finalizare, de aceea vor fi luate în considerare:

- corectitudinea output-ului;
- timpul de răspuns al aplicației (influențat în mare parte de modul de parsare al XML-ului și de calcularea pozițiilor blocurilor în imagine);

### 6.1. Componente critice

Accentul va fi pus pe corectitudinea output-ului, timpul de răspuns în situații critice (ex: imagini deformate) și “corner case”-uri apărute pe parcursul dezvoltării.

Orice element pe care s-a insistat sau de care dezvoltatorul nu este sigur va fi supus verificării.