

**UNIVERSITATEA DIN BUCUREȘTI**



**FACULTATEA DE  
MATEMATICĂ ȘI  
INFORMATICĂ**

**SPECIALIZAREA INFORMATICĂ**

**Lucrare de licență**

**Caiet digital minimalist pentru educație  
pe dispozitive EPD, cu stocare în cloud  
(prototip Evonote)**

**Absolvent**

**Laza George**

**Coordonator științific**

**Lect.dr. Silviu Laurențiu Vasile**

**București, iunie 2025**

## **Rezumat**

Lucrarea de față abordează digitalizarea activității de luare a notițelor la clasă, printr-o soluție minimalistă și accesibilă, concepută special pentru utilizarea în școli. Soluția propusă — EVONOTE — constă într-o aplicație Android care rulează pe o tabletă cu ecran electroforetic (EPD), având funcționalități reduse la strictul necesar, sincronizare automată în cloud și interfață web pentru accesarea și descărcarea notițelor. Scrierea se realizează direct pe suprafața ecranului, cu o latență foarte redusă, printr-un mecanism specific tehnologiei EPD, care pare să evite etapele clasice ale procesării grafice. Tableta este concepută ca sistem închis, pentru a elimina distragerile și a încuraja concentrarea. Întregul sistem este optimizat pentru a funcționa pe un hardware modest, ceea ce permite fabricarea unor dispozitive ieftine, cu autonomie extinsă și costuri reduse de întreținere. EVONOTE își propune să ofere o alternativă digitală simplă, economică și realistă la caietele clasice, accesibilă oricărui elev, contribuind astfel la protecția vederii, reducerea greutății ghiozdanului, conservarea mediului și arhivarea digitală a notițelor pe termen lung.

## **Abstract**

This paper addresses the digitalization of classroom note-taking through a minimalist and accessible solution, specifically designed for use in schools. The proposed system — EVONOTE — consists of an Android application running on a tablet with an electrophoretic display (EPD). It provides only the essential functionality, featuring automatic cloud synchronization and a web interface for accessing and downloading notes. Writing takes place directly on the screen surface with very low latency, enabled by a mechanism specific to EPD technology, which appears to bypass the traditional stages of graphical processing. The tablet is designed as a closed system, eliminating distractions and encouraging focus. The entire system is optimized to run on modest hardware, making it possible to produce low-cost devices with extended battery life and minimal maintenance needs. EVONOTE aims to offer a simple, economical, and practical digital alternative to traditional notebooks — accessible to any student — while contributing to eye protection, lighter schoolbags, environmental preservation, and long-term digital archiving of notes.

# Cuprins

<b>1</b>	<b>Introducere</b>	<b>5</b>
<b>2</b>	<b>Preliminarii</b>	<b>7</b>
2.1	Latența de scriere – aspect de interes în scrierea digitală de mână .....	7
2.2	Tehnologia EPD – principii și avantaje .....	9
2.3	Pipeline-ul clasic de procesare (LCD/OLED) .....	11
2.4	Pipeline-ul EPD — afișare directă fără randare grafică intermediară .....	12
2.5	Alte considerații relevante .....	15
2.6	Considerații privind alegerea tehnologiei EPD .....	16
<b>3</b>	<b>Arhitectura EVONOTE</b>	<b>19</b>
3.1	Arhitectura generală .....	19
3.2	Aplicația Android .....	21
3.2.1	Mecanismul de scriere digitală .....	21
3.2.2	Gestionarea datelor .....	21
3.2.3	Clasa principală PageMain .....	21
3.2.4	Interfața aplicației .....	22
3.3	Mecanismul complet de scriere digitală: captare, procesare și afișare .....	22
3.4	Componenta web a sistemului EVONOTE .....	23
3.4.1	Arhitectura aplicației web .....	23
3.4.2	Interfața utilizatorului .....	24
3.4.3	Comunicarea server - tableta: înrolarea .....	24
3.4.4	Comunicarea server - tableta: sincronizarea caietelor / materiilor .....	25
3.4.5	Comunicarea server - tableta: trimiterea scrierii către server .....	25
3.4.6	Recompunerea scrierii .....	26
3.5	Analiza resurselor consumate .....	27
3.5.1	Analiza consumului de memorie heap .....	27
3.5.2	Analiza procesorului (CPU) .....	28
3.5.3	Estimarea spațiului ocupat de datele de scriere .....	28
3.5.4	Concluzie: Impactul total asupra resurselor .....	29
<b>4</b>	<b>Limitări, concluzii și direcții viitoare</b>	<b>31</b>
	<b>Bibliografie</b>	<b>32</b>

<b>Anexe</b>	<b>34</b>
Anexa 1 – Callback-ul <code>getRawInputCallback()</code> .....	34
Anexa 2 – Secvențe de cod semnificative .....	36
Anexa 3 – Imagini și capturi de ecran .....	37

# 1. Introducere

În ultimele decenii, am asistat la o transformare tehnologică accelerată, poate cea mai profundă din istoria umanității. Și sistemul educațional a urmat această direcție generală, cu o observație care privește direct lucrarea de față: **scrisul de mână a rămas de neînlocuit.**

Spre deosebire de tastatură, scrisul de mână joacă un rol fundamental în învățare. Chiar și simplul act de a copia un text contribuie semnificativ la memorarea acestuia. De asemenea, în activitatea profesională — în special în programare — luarea rapidă de notițe, desenarea unor scheme logice sau memorarea unor coduri și denumiri temporare este mult mai eficientă în scris de mână. Mijloacele digitale existente nu reușeau să ofere o alternativă viabilă la hârtie.

O schimbare majoră a avut loc în 2020, odată cu apariția tabletei **reMarkable 2**, unul dintre primele dispozitive care a utilizat tehnologia E-Ink într-un mod optimizat pentru scrisul de mână, oferind o latență extrem de redusă și o experiență aproape identică cu cea de pe hârtie. A fost momentul în care am realizat potențialul uriaș al acestei tehnologii, nu doar pentru uz personal, ci și pentru mediul educațional. Cu toate acestea, dispozitivele comerciale existente s-au dovedit adesea limitative: interfețe încărcate, aplicații inutile, prețuri ridicate și performanțe medii în afara funcției de scriere. Dar printre toate aceste neajunsuri, **ceea ce conta cu adevărat era acolo:** scrisul fără latență, natural, precis.

În loc să căutăm o soluție care să concureze cu tabletele existente pe toate planurile — performanță grafică, multitasking, aplicații — am ales o altă cale: am micșorat problema. Nu am încercat să adaptăm un sistem complex la școală, ci am redus intenționat complexitatea până la esențialul educației: scrisul.

Această alegere reflectă una dintre valorile creative fundamentale ale proiectului EVONOTE: **minimalismul funcțional.** În locul unui produs multifuncțional, am gândit un dispozitiv specializat, care face doar ceea ce este cu adevărat necesar — și o face bine. **Minimalismul** nu este un compromis, ci un principiu activ de design, cu aplicare atât la nivel hardware (procesor și RAM modeste), cât și software (fără aplicații suplimentare, fără distrageri, fără multitasking artificial).

Al doilea principiu esențial care fundamentează EVONOTE este **accesibilitatea**

**reală.** Tehnologia educațională nu are valoare dacă nu poate fi extinsă la scară largă. De aceea, fiecare decizie de design a fost evaluată nu doar tehnic, ci și economic: va putea fi produs acest dispozitiv în zeci sau sute de mii de exemplare? Poate fi susținut de un buget public? Poate fi menținut un preț corect, fără sacrificii de funcționalitate? Dacă răspunsul este da, atunci avem o soluție viabilă pentru viitorul educației.

Din această frustrare constructivă a luat naștere ideea **EVONOTE: un caiet digital minimalist**, fără aplicații sociale, fără email sau mesagerie, dedicat exclusiv scrisului de mână. Dispozitivul ideal ar permite gestionarea mai multor „caiete” digitale, salvarea automată în cloud, blocarea utilizatorului în aplicația educațională și utilizarea unor „file” ce imită dictando-ul sau pătrățelele din matematică.

În urma studiului pieței, am descoperit că doar un singur producător — ONYX BOOX — oferă un SDK funcțional pentru dezvoltarea de aplicații care utilizează scrierea pe ecran EPD, pe un sistem Android. Am decis astfel să folosesc acest ecosistem pentru a implementa prototipul EVONOTE.

Pe lângă beneficiile directe în învățare, soluția propusă adresează și probleme practice și sociale:

- **reducerea greutății ghiozanelor** elevilor (o cauză recunoscută a problemelor de coloană la copiii mici), păstrarea digitală a notițelor pe termen lung,
- **limitarea distragerilor digitale printr-o interfață controlată** — și, nu în ultimul rând,
- **protejarea mediului** prin reducerea masivă a consumului de hârtie,
- **păstrarea digitală a notițelor pe termen lung**, într-o arhivă accesibilă și organizată, care sprijină **incluziunea digitală** prin posibilitatea ca dispozitivul — eficient, robust și cu cost redus — să fie adoptat la scară largă, nu doar de cei privilegiați,
- **protejarea vederii**, datorită ecranului E Ink fără lumină emisivă, care elimină oboseala oculară specifică ecranelor clasice și permite utilizarea îndelungată fără disconfort vizual.

Lucrarea de față documentează procesul de analiză, proiectare și implementare a unui astfel de sistem, sub forma unui prototip funcțional în componentele sale esențiale. Acesta include aplicația Android pentru tabletă cu ecran EPD, serverul de sincronizare și interfața web pentru accesarea notițelor. Deși anumite funcționalități (precum modul kiosk) nu au fost încă implementate, prototipul oferă o dovadă solidă de concept, demonstrând fezabilitatea și potențialul real al proiectului EVONOTE în

context educațional.

În mod esențial, EVONOTE nu este gândit ca un produs de nișă sau ca un accesoriu tehnologic pentru elitele financiare. Dimpotrivă, obiectivul este dezvoltarea unei soluții accesibile, scalabile și suficient de robuste pentru a putea fi adoptată la scară largă în învățământul public. Un caiet digital nu trebuie să fie un lux, ci o unealtă de bază — la fel cum hârtia și pixul sunt astăzi.

În condițiile unei producții la scară largă și a unei arhitecturi software bine calibrate, este plauzibil ca prețul unui astfel de dispozitiv să poată fi redus semnificativ — până la punctul în care fiecare copil să poată avea propriul caiet digital.

Viitorul este al tuturor, iar tehnologia ar trebui să servească întreaga umanitate, nu doar un segment privilegiat. În această viziune se înscrie și EVONOTE — o propunere concretă pentru digitalizarea echitabilă a educației.

## **2. Preliminarii**

### **2.1. Latența de scriere – aspect de interes în scrierea digitală de mână**

În interacțiunea om-mașină, latența de scriere reprezintă intervalul de timp dintre acțiunea utilizatorului și reacția vizibilă a sistemului. În contextul scrisului de mână digital, orice întârziere perceptibilă între mișcarea stylus-ului și apariția efectivă a urmei pe ecran poate rupe fluiditatea și naturalețea gestului.

În contextul acestei lucrări, prin latență de scriere înțelegem timpul necesar pentru întregul proces pornind de la input-ul utilizatorului la procesarea semnalului și în final la afișarea pe ecran care nu poate fi mai devreme de următorul ciclu de refresh.

Din punct de vedere perceptiv, studiile indică faptul că limita inferioară la care creierul uman poate detecta o întârziere se situează în jurul valorii de 13 ms [1]. Totuși, în contextul vizual, toleranța este mult mai mare: experiența cinematografică tradițională funcționează la 24–29 de cadre pe secundă, ceea ce înseamnă un interval între cadre de 34–41 ms — suficient pentru ca mișcarea să fie percepută ca fluidă. Scrisul, însă, este un proces activ și bidirecțional, ceea ce înseamnă că latențele devin sesizabile și chiar deranjante la valori mult mai mici.

Această problemă este ilustrată clar într-un clip video publicat în martie 2012 de Microsoft Applied Sciences Group, unde Paul Dietz explorează efectele latenței asupra percepției utilizatorului [2]. În cadrul acestui experiment, un dreptunghi este tras pe ecran cu degetul, iar imaginea sa este întârziată progresiv pentru a simula diverse niveluri de latență:

- La 100 ms, latența era specifică majorității dispozitivelor de atunci și extrem de vizibilă.
- La 50 ms, întârzierea devine deranjantă în utilizarea reală.
- La 10 ms, obiectul „iese de sub deget” în timpul schimbărilor bruște de direcție.
- Abia la 1 ms, mișcarea este perfect sincronizată cu input-ul și creează impresia unui obiect fizic real.

Scrisul digital, deși implică mișcări ceva mai lente și distanțe mai mici decât glisarea unui obiect, este supus aceluiași limitări de percepție.

Dispozitivele moderne high-end au redus semnificativ această latență. De exemplu, o tabletă **Samsung Galaxy Tab S6 Lite** are un refresh rate de 60 Hz, ceea ce înseamnă o distanță de 16,67 ms între cadre. Aceasta este limita teoretică minimă pentru afișare, dar în practică trebuie adăugat și timpul de procesare grafică și cel de randare, astfel că latența reală rămâne deseori peste 20–30 ms.

Totuși, într-o recenzie a dispozitivului se afirmă „*there is very little lag*” (există foarte puțină întârziere) atunci când se preintă capacitățile de scriere [3]. Mai mult, într-o discuție pe reddit între utilizatori despre acest dispozitiv e publicat un videoclip filmat la 240fps în care linia urmează traiectoria creionului la 10-12 frame-uri distanță, deci manifestă o latență de 42-50 ms [4].

În cazul **Lenovo P11 Pro** cu **Precision Pen 2** o recenzie dedicată afirmă: „*There's input lag with all the drawing apps. Lines will try to catch up to the pen tip. That's just the way it is with 60Hz display.*” (există latență în toate aplicațiile de desen; liniile încearcă să prindă din urmă vârful stylusului; acesta este comportamentul tipic al unui ecran de 60Hz) [5].

Nici iPad nu a scăpat de recenzii critice, iar pe forumul oficial Apple (discussions.apple.com) este semnalată o **altă sursă de latență**, diferită de limita impusă de rata de refresh a ecranului (ex. 60Hz). În acest caz, întârzierea este explicată prin lipsa resurselor sistemului: „... latency (i.e, slow or delayed response) is often

attributable to lack of system resources for the task being performed - either due to lack of working memory (RAM) or too many other processes running in the background” (întârzierea este adesea cauzată de lipsa resurselor de sistem necesare sarcinii — fie din cauza unei memorii RAM insuficiente, fie din cauza prea multor procese care rulează în fundal) [6].

În concluzie, latența de scriere rămâne unul dintre cele mai importante criterii în alegerea unui dispozitiv digital care să înlocuiască hârtia. Doar un sistem care reușește să reducă această întârziere sub pragul perceptibil (ideal sub 30 ms) poate oferi o experiență autentică de scriere — iar acest obiectiv a fost mult timp greu de atins cu tehnologiile de afișare convenționale precum LCD sau OLED.

Chiar și în prezent, atingerea acestui nivel de performanță implică utilizarea unor dispozitive cu hardware de ultimă generație, în combinație cu o gestiune atentă a resurselor — fără aplicații active în fundal, fără multitasking agresiv și cu optimizări la nivel de sistem. În lipsa acestor condiții, latența reappare chiar și pe cele mai moderne tablete.

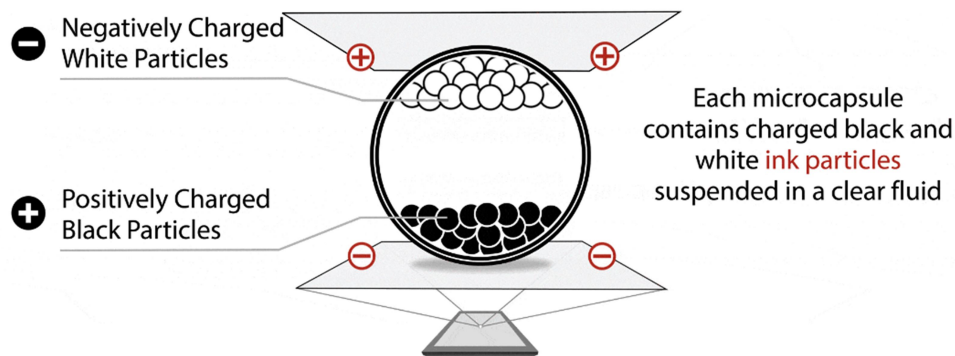
O alternativă interesantă a apărut în 2020, odată cu lansarea tabletei reMarkable 2, care utilizează un ecran EPD (Electrophoretic Display) și oferă o latență declarată de doar 21 ms în modul de scriere. La momentul respectiv, majoritatea tabletelor din gama medie prezentau în continuare întârzieri notabile în afișarea traseului stylus-ului, ceea ce afecta semnificativ experiența utilizatorului [7].

Tabletele reMarkable, la fel ca alte dispozitive bazate pe tehnologie EPD, oferă o senzație de scriere naturală aproape perfectă, nu doar datorită latenței reduse, ci și caracteristicilor fizice ale ecranului: suprafață complet mată, fără reflexii și neemitentă de lumină, chiar și în configurațiile care includ iluminare frontală. În plus, consumul energetic extrem de redus al tehnologiei E-Ink permite o autonomie remarcabilă — de până la două săptămâni de utilizare cu o baterie de numai 3000 mAh.

## **2.2. Tehnologia EPD – principii și avantaje**

Tehnologia EPD (Electrophoretic Display), cunoscută și sub denumirea de „e-ink” sau „ePaper”, reprezintă o alternativă fundamental diferită față de afișajele clasice LCD sau OLED. În loc să emită lumină, un ecran EPD reflectă lumina ambientală în mod pasiv, imitând astfel aspectul unei pagini tipărite. Această caracteristică îl face

ideal pentru citit și scris, mai ales în condiții de iluminare naturală.



**Figura 2.1:** Principiul de funcționare al unei microcapsule E Ink. Particulele albe încărcate pozitiv și cele negre încărcate negativ sunt suspendate într-un fluid transparent. Sub influența câmpului electric, ele migrează spre electrozii de polaritate opusă, determinând dacă pixelul va apărea alb sau negru la suprafață [8].

La nivel tehnic, un EPD este compus din milioane de microcapsule suspendate într-un lichid transparent. Fiecare microcapsulă conține particule albe încărcate pozitiv și particule negre încărcate negativ. Prin aplicarea unui câmp electric între doi electrozi (sus și jos), particulele se deplasează în funcție de polaritate: cele albe urcă la suprafață, afișând alb, iar cele negre coboară, sau invers, determinând astfel culoarea vizibilă în acea zonă.

Această metodă permite o afișare extrem de stabilă — imaginea rămâne vizibilă fără consum de energie până când este modificată — și foarte eficientă energetic, întrucât energia este consumată doar la schimbarea conținutului vizual. De asemenea, lipsa iluminării din spate și a emisiei de lumină albastră contribuie la reducerea oboselei oculare, un avantaj major pentru utilizarea în educație sau în medii unde cititul și scrisul se fac pe durate îndelungate [8].

#### **Notă privind percepția afișării pe EPD:**

Deși scrierea pe un ecran EPD poate părea instantanee, este important de înțeles că imaginea nu apare brusc, ca pe un LCD, ci se formează progresiv. Procesul electrofizic prin care pigmentii se rearanjează în microcapsule are o durată detectabilă — chiar dacă primele urme ale liniei apar extrem de repede, iar ochiul uman este „păcălit” să perceapă mișcarea ca fiind continuă. Acest comportament este specific afișajelor EPD, unde pigmentii nu comută instantaneu, ci migrează treptat în fluidul din microcapsule,

rezultând o apariție progresivă a imaginii.

Fenomenul este confirmat și vizual: o filmare proprie inclusă în această lucrare evidențiază clar modul în care linia scrisă se formează gradual, cadru cu cadru, chiar și la o rată de 30 fps (vezi Figura 2.3).

## 2.3. Pipeline-ul clasic de procesare (LCD/OLED)

Pe dispozitivele Android convenționale, scrierea digitală implică traversarea unui lanț complex de componente hardware și software, fiecare contribuind cu o întârziere specifică. Acest proces, cunoscut sub numele de pipeline de afișare, este alcătuit din mai multe etape secvențiale, începând de la captarea semnalului stylus-ului și terminând cu afișarea efectivă pe ecran.

### **Procesul poate fi rezumat astfel:**

a. Captarea inputului: Stylus-ul generează semnalul de scriere, care este preluat de digitizer (stratul senzorial al ecranului), colectând date precum coordonatele (x, y), presiunea și uneori unghiul de înclinare. Aceste informații sunt procesate și convertite în evenimente de tip MotionEvent [9].

Latență estimată: 2-7 ms – digitizerele moderne sunt rapide și pot lucra la frecvențe de 240Hz, chiar și în aceste condiții, așa un ciclu înseamnă 4.17ms.

b. Transmiterea către sistemul de operare: Evenimentele captate sunt preluate de componentele interne Android InputReader și InputDispatcher, care le plasează într-o coadă de evenimente și le transmit aplicației active [10].

Latență estimată: 2-5 ms – chiar dacă aceste operațiuni sunt optimizate, coada de evenimente și procesarea asincronă pot introduce mici întârzieri, în funcție de încărcarea sistemului și prioritizarea thread-urilor.

c. Prelucrarea în aplicație: Aplicația interpretează datele primite și determină acțiunea necesară – de exemplu, desenarea unei linii, alegerea culorii sau grosimii acesteia.

Latență estimată: 2-10 ms – timpul variază în funcție de logica aplicației, complexitatea codului și performanța procesorului. Aplicațiile mai simple reacționează mai rapid, însă aplicațiile bogate în funcționalități pot introduce latențe suplimentare.

d. Randarea grafică: Instrucțiunile de desenare sunt transmise către procesorul grafic (GPU), care generează imaginea într-un buffer de memorie (frame buffer).

Latență estimată: 4-10 ms – timpul necesar depinde de complexitatea vizuală (straturi, transparențe, efecte), dar și de eficiența pipeline-ului grafic și de încărcarea GPU-ului în acel moment [10].

e. Sincronizarea cu afișajul: Android utilizează semnalul VSYNC pentru a sincroniza afișarea cu rata fixă de reîmprospătare a ecranului. La o frecvență standard de 60 Hz, un cadru nou poate fi afișat doar o dată la 16.67 ms, ceea ce înseamnă că imaginea trebuie adesea să aștepte următorul ciclu disponibil [11].

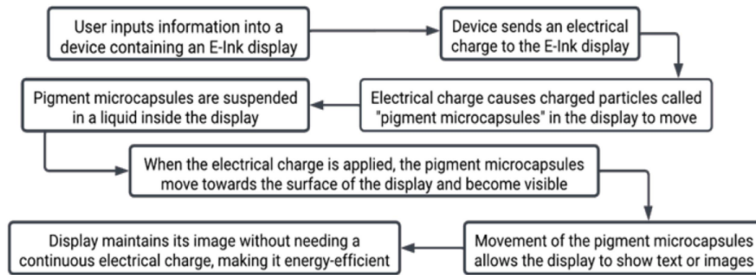
Latență estimată: 0–16.67 ms – în funcție de momentul în care bufferul este gata față de semnalul VSYNC, întârzierea poate varia între zero și întregul ciclu de refresh.

Latența totală rezultată din acest lanț de procesare se situează, în practică, între 25 și 60 ms. Această întârziere este suficientă pentru a fi percepută de utilizator sub forma unui decalaj vizibil între mișcarea stylus-ului și apariția liniei — fenomen cunoscut sub numele de lag. Chiar dacă poate fi redus cu hardware performant, acest lucru implică un cost și un consum energetic ridicat, greu de justificat în contextul educației de masă.

## **2.4. Pipeline-ul EPD — afișare directă fără randare grafică intermediară**

Spre deosebire de ecranele LCD sau OLED, afișajele EPD (Electrophoretic Display) permit o abordare fundamental diferită în procesarea scrierii de mână. Imaginea nu este generată printr-un proces grafic clasic, ci direct în stratul fizic al ecranului — prin rearanjarea locală a particulelor din microcapsulele de cerneală electronică, ca reacție la un câmp electrostatic aplicat punctual.

Funcționarea de bază a afișajului EPD este ilustrată în Figura 2.2, care sintetizează modul în care inputul utilizatorului duce la aplicarea unui câmp electric asupra microcapsulelor de pigment suspendate într-un lichid. Particulele încărcate electric migrează către suprafață, devenind vizibile și formând imaginea fără implicarea unui GPU, a unui buffer grafic sau a sincronizării prin VSYNC [12].



**Figura 2.2:** Pipeline-ul fizic și hardware-centric al unui ecran EPD [12].

În cadrul aplicației EVONOTE, acest mecanism este accesat prin intermediul SDK-ului oferit de BOOX, care include o funcționalitate neoficial denumită RawDrawing, așa cum reiese din numele metodelor utilizate în exemplele de cod furnizate de producător (ex: `openRawDrawing()`, `setRawDrawingEnabled(true)` etc.) [13].

Activarea acestui mod permite preluarea în timp real a input-ului stylus-ului și afișarea directă a liniei scrise, fără intervenția procesorului grafic (GPU), fără utilizarea unui buffer grafic și fără sincronizare cu frecvența de reîmprospătare a ecranului. Scrisul devine astfel instantaneu, fluid și vizibil exact sub vârful stylus-ului, oferind o experiență foarte apropiată de cea a scrisului pe hârtie.

Pentru a investiga empiric comportamentul afișajului EPD în modul RawDrawing, a fost realizat un experiment simplu de filmare cu un telefon mobil, urmat de analiză cadru-cu-cadru folosind VLC. În secvența surprinsă (vezi Figura 2.3), se pot observa clar următoarele:

- scrierea începe aproape imediat după contactul stylus-ului cu ecranul;
- linia se formează treptat, în mod vizibil progresiv;
- după aproximativ un frame ( $\approx 33$  ms, la 30 cadre/secundă în filmarea analizată), linia este deja conturată într-o proporție mare (peste 75%), iar în cadrul următor este completă.



**Figura 2.3:** Cinci cadre succesive extrase dintr-o filmare proprie realizată la 30 fps pe un dispozitiv BOOX Note 2. Se observă, pe de o parte, o latență de aproximativ un cadru ( $\approx 33$  ms), iar pe de altă parte, formarea treptată a liniei scrise.

O observație empirică suplimentară vine din comportamentul unei alte tablete testate, Likebook Muse de 7", care nu folosește SDK-ul BOOX dar manifestă o logică similară. În timpul scrierii, imaginea apare instantaneu, direct pe ecran. În schimb, la ștergere, sistemul nu poate modifica local zona afectată, ci reconstruiește vizual pagina din datele stocate: coordonate, presiune, stil de linie etc. Diferența este vizibilă cu ochiul liber: imaginea redesenată este ușor diferită față de versiunea scrisă inițial, ceea ce confirmă existența unui mecanism dual: scriere fizică directă și reconstrucție logică.

Un aspect esențial care susține aceste observații este diferența de comportament între scriere și ștergere în aplicația EVONOTE. Scrierea cu stylus-ul utilizează modul RawDrawing și, conform comportamentului observat și analizei SDK, nu pare să implice apeluri directe către canvas sau GPU. În schimb, reafișarea paginii după ștergere folosește metoda:

```
EpdController.enablePost(surfaceView, 1);
```

Acest apel activează zona de ecran în modul DU (Direct Update) — un mod rapid de reîmprospătare locală specific ecranelor E Ink. Acest proces este mai lent și vizibil diferit, confirmând existența a două căi de afișare:

- una fizică, directă, pentru scrierea nouă;
- una logică, clasică, pentru reafișare sau modificări.

Trebuie menționat și că documentația tehnică referitoare la aceste mecanisme este extrem de limitată. De exemplu, nici măcar controllerul folosit de BOOX Note 2 nu

este documentat public. Pentru a clarifica aceste aspecte, a fost transmisă o solicitare directă producătorului ecranului (E Ink), dar până la data redactării nu a fost primit niciun răspuns. Accesul la astfel de informații ar necesita demontarea dispozitivului și analiză hardware avansată, ceea ce depășește scopul acestei lucrări. În acest context, ipotezele formulate se bazează pe o combinație de analiză experimentală, comportament observabil, cod sursă și raționament tehnic.

Această arhitectură unică are implicații importante asupra cerințelor hardware: tabletele EPD pot funcționa fluent cu procesoare modeste, fără GPU și cu consum minim de memorie. Eliminarea etapelor clasice de randare duce la un consum redus de energie și la posibilitatea construirii unui dispozitiv educațional ieftin, eficient și sustenabil — exact ceea ce își propune proiectul EVONOTE.

## 2.5. Alte considerații relevante

Deși atinse în treacăt în secțiunile anterioare, există câteva aspecte importante care merită punctate separat, deoarece întăresc alegerea tehnologiei EPD în contextul unei aplicații educaționale durabile, accesibile și prietenoase cu utilizatorul.

**Protecția ochilor** – afișajele EPD nu emit lumină proprie, ci reflectă lumina ambientală, asemenea unei pagini tipărite. Acest lucru elimină complet lumina albastră, flicker-ul și alte efecte vizuale dăunătoare specifice ecranelor LCD sau OLED. Pentru utilizarea intensivă în medii educaționale – unde elevii pot privi ore în șir către dispozitiv – acest avantaj devine esențial, contribuind la protejarea sănătății ochilor pe termen lung. Un studiu realizat la Harvard arată că ecranele ePaper (precum E Ink) sunt de până la trei ori mai sănătoase pentru ochi comparativ cu ecranele LCD, în condiții de utilizare prelungită [14].

**Autonomie extinsă** – dispozitivele EPD consumă energie doar atunci când imaginea este modificată. Astfel, în absența animațiilor sau a conținutului video, o tabletă cu o baterie de 3000–4000 mAh poate funcționa până la două săptămâni între încărcări [15]. Această autonomie excepțională este ideală pentru mediul educațional, unde accesul la prize sau infrastructură electrică nu este întotdeauna garantat. Vom reveni asupra acestui aspect după testarea prototipului și stabilirea configurației hardware minime necesare.

**Importanța scrisului de mână în educație** – numeroase studii din psihologie și

pedagogie confirmă că scrisul de mână contribuie semnificativ la memorare, înțelegere și structurarea logică a informației. Actul scrierii implică o procesare activă a conținutului, iar notațiile manuale stimulează simultan coordonarea motorie și cogniția [16]. În acest context, un dispozitiv digital care oferă o experiență de scriere naturală, apropiată de cea pe hârtie, nu este doar un înlocuitor comod, ci un sprijin real pentru procesul educațional.

**Naturaletă percepută a scrisului** – scrisul pe un ecran EPD oferă o experiență extrem de apropiată de hârtie: suprafața mată, lipsa reflexiilor și feedback-ul vizual instantaneu contribuie la o senzație autentică. Această naturaletă devine cu atât mai valoroasă în contextul rolului esențial pe care scrisul de mână îl joacă în procesul de învățare, mai ales în clasele primare [17].

**Necesitate hardware redusă** – pipeline-ul simplificat al tehnologiei EPD permite funcționarea fluentă chiar și pe hardware modest: procesoare de generații mai vechi, fără GPU puternic, cu memorie RAM redusă. Acest lucru reduce costurile de producție și mentenanță și crește durata de viață a dispozitivelor, contribuind la sustenabilitate economică și tehnologică. Confirmarea acestor estimări se va face în cadrul capitolului dedicat testării aplicației și monitorizării resurselor.

## **2.6. Considerații privind alegerea tehnologiei EPD**

Într-un context în care tehnologiile LCD și OLED domină piața tabletelor, alegerea unei tehnologii mai puțin răspândite precum EPD poate părea contraintuitivă. Totuși, dincolo de simpla comparație de performanță brută, EPD-ul oferă un set unic de caracteristici care îl recomandă pentru dezvoltarea unui caiet digital educațional, în special atunci când scopul este accesibilitatea la scară largă, sustenabilitatea și calitatea interacțiunii vizuale.

Alegerea EPD – o opțiune strategică Piața actuală a tabletelor este dominată de dispozitive puternice, cu ecrane performante, funcționalități multimedia și aplicații multiple. În 2025, prețurile între tabletele clasice și cele cu EPD tind să fie similare, însă diferențele în utilizare și scop pot face toată diferența. EPD-ul consumă energie doar la schimbarea imaginii, are autonomie de zile sau chiar săptămâni și nu necesită RAM sau procesor performant. Aceste caracteristici permit reducerea drastică a costurilor, atâta timp cât renunțăm la aplicațiile de fundal și funcționalitățile inutile

care consumă resurse, păstrând doar componentele strict necesare pentru scriere și sincronizare periodică în cloud.

### **De ce nu o tabletă convențională blocată?**

O posibilă alternativă ar fi utilizarea unei tablete Android convenționale cu acces restricționat la aplicații. Totuși, această abordare este suboptimă din mai multe motive:

- Prețul: Tabletele clasice au costuri ridicate pentru ecran, procesor și memorie.
- Irosirea resurselor: Se folosește doar o fracțiune din capabilități, restul fiind inutil.
- Consumul energetic: Frecvența mare de încărcare nu este practică pentru mediul educațional.
- Complexitate și risc: Blocarea software poate fi evitată de copii, iar administrarea devine dificilă.
- Mesaj educațional confuz: Oferi un dispozitiv complet, dar interzis – un contrast care stimulează tentația de a-l debloca.
- Procesele de fundal: Chiar și în lipsa aplicațiilor vizibile, sistemul Android rulează numeroase servicii de fundal (actualizări, notificări, module Google etc.), care consumă resurse și energie.

### **Un sistem închis, fără distrageri**

Tot mai mulți părinți limitează expunerea copiilor la ecrane („screen time”) și întârzie accesul la smartphone. Un dispozitiv educațional ar trebui să reflecte acest trend: fără Google Play, fără aplicații, fără distrageri. EVONOTE este gândit ca sistem dedicat, preconfigurat, fără acces la aplicații sau magazin Play – fiind astfel gata de utilizare fără ajustări suplimentare.

### **Potențialul nevalorificat al EPD**

Tabletele EPD existente sunt concepute ca produse de nișă pentru profesioniști. Ele includ procesoare rapide, conectivitate avansată și multe funcții care nu sunt necesare în acest caz. În schimb, un dispozitiv minimalist, cu o singură funcție — scrisul de mână — poate reduce costul, riscul și complexitatea. Scrierea directă pe ecran EPD elimină nevoia de randare complexă sau accelerare grafică.

### **Eficiența economică a fabricării în serie**

Prețul ridicat al tabletelor EPD este rezultatul lipsei unei producții specializate în masă. Cu o strategie publică sau prin parteneriate educaționale, costul poate fi redus

semnificativ. Dacă hardware-ul este optimizat iar producția este susținută de stat sau industrie, un astfel de dispozitiv poate deveni realitate pentru fiecare elev.

### **Argumente funcționale**

Se mențin toate beneficiile cunoscute ale EPD:

- protejarea sănătății vizuale;
- durata extinsă a bateriei;
- senzația naturală a scrisului de mână;
- cerințele hardware scăzute;
- compatibilitatea perfectă cu arhivarea digitală în cloud.

### **Minimalismul ca element central**

Dacă privim EVONOTE prin prisma minimalismului – concept creativ fundamental al întregului proiect – observăm că soluția EPD este nu doar adecvată, ci optimă. Acest tip de ecran susține ideea unui dispozitiv axat exclusiv pe scris, fără elemente care să-i complice funcționarea sau să-i dubleze costurile.

Această coerență între scop și mijloace se manifestă în două dimensiuni esențiale:

- Hardware simplificat, fără putere de calcul excesivă, care reduce semnificativ costul per unitate.
- Software redus la esențial, fără aplicații sau funcții parazite, ceea ce favorizează concentrarea și disciplina digitală.

Această simplitate nu este o limitare, ci o alegere conștientă – una care transformă minimalismul într-un atu educațional, economic și tehnologic.

Concluzie Alegerea EPD nu este un moft tehnologic, ci o alegere strategică. Este singura tehnologie care permite construirea unui sistem educațional digital cu adevărat simplu, accesibil, durabil și scalabil. Este adaptată nevoilor reale ale educației moderne și în acord cu cerințele sociale legate de sănătate, protecția copiilor și sustenabilitate.

Această alegere vine cu provocări reale. Trebuie luate în calcul inerția pieței, prejudecățile tehnologice și, uneori, reticența instituțională. Însă tocmai în fața acestor obstacole, o soluție clară, coerentă și sănătoasă își dovedește valoarea. Dacă vom reuși să o implementăm, impactul pozitiv ar putea fi semnificativ. Sau, parafrazându-l pe John F. Kennedy: alegem această direcție nu pentru că este ușoară, ci pentru că este corectă.

# 3. Arhitectura EVONOTE

## 3.1. Arhitectura generală

Sistemul EVONOTE este conceput ca o platformă distribuită, formată din trei componente esențiale: aplicația de pe tabletă (dispozitiv EPD), serverul cloud și interfața web de acces. Împreună, acestea formează un ecosistem educațional coerent, axat pe scrisul de mână, sincronizare automată și arhivare digitală pe termen lung.

### Arhitectura generală (tabletă – cloud – web)

Sistemul permite conectarea a mai multor utilizatori, fiecare cu unul sau mai multe dispozitive.

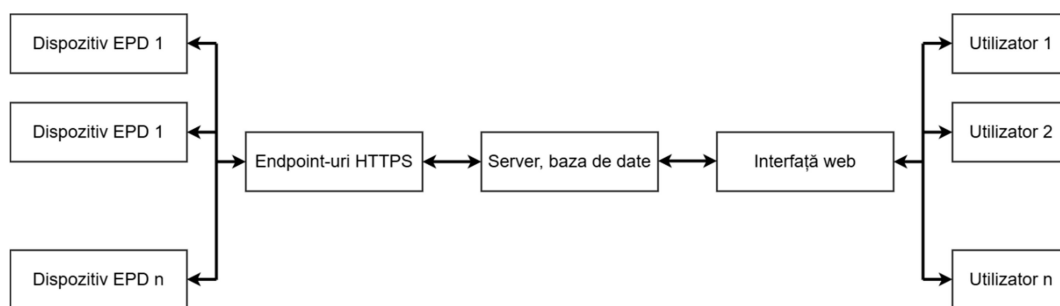


Figura 3.1: Arhitectura generală a sistemului EVONOTE.

### Procesul de inițializare

La prima utilizare a tabletei, aplicația parcurge un proces de inițializare în trei pași:

- **Crearea contului de utilizator** pe platforma web EVONOTE;
- **Înrolarea tabletei**, cu generarea locală a unei chei unice (dkey), asociată contului și utilizată ulterior pentru autentificare și identificare.
- **Definirea caietelor** pe platforma web: fiecare caiet are un IDbook, o materie, un an școlar etc., iar aplicația le preia și le salvează local pe dispozitiv

### Fluxul informațiilor și sincronizarea

- Tableta înregistrează datele scrise în baza locală SQLite.
- La anumite intervale sau la cerere, datele sunt transmise către server în format JSON, prin conexiune securizată HTTPS.
- Fiecare mesaj include:
  - cheia dkey, Idbook

- datele de scris (coordonate, presiune, timp),
- marcaje pentru întreruperi (list\_stop),
- tipul de operație (scriere / ștergere).
- Serverul validează cu dkey, stochează datele și răspunde cu un cod de confirmare în baza căruia aplicația marchează local datele ca sincronizate.

### **Accesul utilizatorului la conținutul scris**

După ce datele au fost sincronizate cu serverul, utilizatorul le poate accesa prin intermediul interfeței web EVONOTE. Platforma permite:

- navigarea prin caiete, pagină cu pagină;
- vizualizarea scrierii în format grafic;
- descărcarea caietelor în format PDF.

Această funcționalitate asigură o arhivare pe termen lung și permite accesul facil la notițe de pe orice dispozitiv, fără a depinde de tableta originală.

### **Actualizarea definițiilor**

Aplicația permite actualizarea manuală a listei de caiete printr-un buton dedicat. Utilizatorul folosește această opțiune atunci când, de exemplu, a adăugat sau modificat un caiet în platforma web.

- Caietele nou definite sunt preluate local la cerere.
- Modificările de denumire sau structură devin vizibile după actualizare.
- Caietele eliminate pe server devin inactive pe tabletă.

Sincronizarea conținutului scris are loc doar într-un singur sens: de la tabletă către server.

### **Suport multi-user și multi-device**

Sistemul EVONOTE este conceput să funcționeze în contexte educaționale reale, cu mai mulți utilizatori și dispozitive:

- Un utilizator poate folosi mai multe tablete, fiecare cu propriile;
- Fiecare tabletă este asociată unui singur utilizator, încă de la înrolare;
- Mai mulți utilizatori sunt gestionați complet separat, fiecare cu propriul set de tablete și caiete.

Arhitectura este proiectată să fie **ușor scalabilă**, cu posibilitate de extindere pentru volume mari (zeci sau sute de mii de utilizatori), prin adaptări ulterioare ale infrastructurii hardware și ale sistemului software.

## **3.2. Aplicația Android**

Aplicația EVONOTE a fost dezvoltată folosind Android SDK 9.0, în Java, având ca bază funcționalitățile oferite de SDK-ul oficial Onyx [13] pentru tableta BOOX Note 2. Implementarea are la bază aproximativ zece clase proprii, organizate clar în jurul activității principale PageMain, unde se gestionează scrierea, interacțiunea cu stylusul, paginarea și salvarea.

### **3.2.1. Mecanismul de scriere digitală**

Acesta presupune captarea inputului de la stylus, salvarea datelor și redarea grafică pe ecran cu latență minimă, reprezintă o componentă esențială a aplicației EVONOTE. Datorită complexității sale, acest proces este tratat detaliat într-un capitol separat (3.3), unde sunt prezentate atât structura logică a prelucrării punctelor, cât și interacțiunea cu sistemul de afișare EPD specific tabletelor Onyx BOOX.

### **3.2.2. Gestionarea datelor**

Toate punctele scrise sunt salvate local într-o bază de date SQLite, folosind o clasă dedicată DBHelper. Structura unui punct este [x|y|presiune|timp|z], salvată în câmpul list\_pts. De asemenea, sunt marcate pauzele (pentru a nu crea linii continue între segmente) și tipul de acțiune (type=1 scriere, type=-1 ștergere).

O parte importantă a funcționalității aplicației constă în sincronizarea acestor date cu serverul. La comanda utilizatorului, aplicația serializează punctele în format JSON și le trimite prin HTTPS către server, cu o cheie de autentificare (dkey), iar răspunsul serverului este procesat pentru a actualiza progresul salvării.

### **3.2.3. Clasa principală PageMain**

Logica esențială a aplicației se află în clasa PageMain, unde sunt definite:

- inițializarea suprafeței de scris (SurfaceView) și activarea desenului;
- gestionarea paginilor: crearea, navigarea, selectarea;
- comanda de salvare în cloud și afișarea progresului;
- selectarea grosimii liniei printr-un set de butoane vizuale;
- dialogul de alegere pagină și adăugare de pagini noi.

Toate componentele sunt gândite pentru o utilizare intuitivă și concentrată pe scopul

educațional al aplicației.

### 3.2.4. Interfața aplicației

Interfața EVONOTE a fost concepută minimalist, urmărind simplitatea și claritatea necesare unui instrument educațional. Butoanele de interacțiune sunt fixe, plasate în partea superioară a ecranului, și includ: navigarea între pagini, selectarea grosimii liniei, alegerea unei pagini din listă, salvarea conținutului și închiderea aplicației. Toate funcțiile esențiale sunt accesibile direct, fără meniuri ascunse sau interacțiuni complexe, tocmai pentru a susține ideea de „caiet digital” utilizabil zilnic în mediul școlar.

În cadrul **Anexei 3** sunt incluse capturi de ecran care ilustrează interfața și componentele principale ale aplicației.

## 3.3. Mecanismul complet de scriere digitală: captare, procesare și afișare

Pentru captarea scrisului de mână în timp real, a fost utilizată clasa TouchHelper din SDK-ul Onyx, care permite preluarea brută a input-ului stylus-ului prin funcția `getRawInputCallback()`.

```
touchHelper = TouchHelper.create(surfaceView, getRawInputCallback());  
touchHelper.openRawDrawing();  
touchHelper.setRawDrawingEnabled(true);
```

Această secvență activează modul RawDrawing, permițând scrierea directă pe suprafața EPD, fără a aștepta pipeline-ul grafic standard Android. Stylusul devine practic o „cerneală digitală” cu latență imperceptibilă.

Callback-ul `getRawInputCallback()` este responsabil pentru captarea în timp real a punctelor desenate sau șterse de stylus pe ecran. La fiecare interacțiune, sunt preluate coordonatele (x, y), presiunea, dimensiunea punctului și momentul temporal (timestamp). Aceste date sunt salvate local într-o bază SQLite și, ulterior, sincronizate în cloud.

În cadrul acestui callback sunt gestionate toate etapele scrierii și ștergerii, de la inițiere până la finalizare. O selecție relevantă de metode și fragmente de cod ilustrative este prezentată în Anexele 1 și 2.

La fiecare gest de scriere sau ștergere, aplicația preia datele de la stylus prin RawInputCallback, le salvează logic într-o structură temporară și apoi le salvează și în baza de date.

În paralel, la finalurile de linie, punctele sunt desenate pe un canvas din memorie (renderBitmap), iar imaginea este actualizată pe ecran prin metoda drawBitmap(), utilizând funcția EpdController.enablePost() pentru afișare instantanee în mod DU (Direct Update).

Acest mecanism evidențiază modul dual de procesare a inputului oferit de hardware-ul EPD: afișare imediată prin RawDrawing, combinată cu înregistrarea detaliată a punctelor pentru redarea ulterioară a imaginii. Abordarea este inspirată din proiectele demonstrative puse la dispoziție de producător pe GitHub, care includ atât componente SDK, cât și exemple funcționale de aplicații Android dedicate scrisului pe ecrane E Ink.

## **3.4. Componenta web a sistemului EVONOTE**

Sistemul EVONOTE nu se limitează la aplicația de pe tableta, ci include și o platformă web complementară, al cărei rol este să ofere sincronizare, stocare și acces pe termen lung la notitele elevilor. În acest capitol sunt detaliate componentele esențiale ale acestei infrastructuri.

### **3.4.1. Arhitectura aplicației web**

Aplicația web EVONOTE este construită utilizând PHP pentru partea de backend, MySQL pentru gestionarea bazei de date și HTML+CSS pentru interfața cu utilizatorul. Arhitectura adoptată este una tradițională, bazată pe generarea server-side a paginilor HTML și o navigare secvențială, pagină cu pagină. În cazurile care necesită o interactivitate sporită, sunt utilizate apeluri AJAX punctuale.

Aplicația web a fost construită pe baza unui CMS propriu, care include funcționalități standard (gestionarea utilizatorilor) și componente personalizate pentru administrarea dispozitivelor, caietelor digitale și a conținutului asociat. Totuși, aceste funcționalități țin mai degrabă de infrastructura generală și nu fac obiectul central al acestei lucrări — ele sunt menționate aici doar pentru a contura cadrul tehnologic în care se încadrează sistemul EVONOTE.

### **3.4.2. Interfața utilizatorului**

Interfața web a platformei EVONOTE oferă utilizatorilor (în principal elevi și, opțional, părinți sau titori) acces direct și organizat la notițele sincronizate de pe tabletă. Aceasta permite:

- administrarea anilor școlari, a dispozitivelor (tablete) și a caietelor digitale asociate materiilor;
- navigarea secvențială, pagină cu pagină, în cadrul fiecărui caiet sau materie;
- descărcarea integrală a oricărui caiet în format PDF;
- efectuarea tuturor acțiunilor specifice gestionării unui cont individual de utilizator (ex. schimbare parolă, resetare date, revizuire dispozitive asociate).

Platforma este concepută pentru a permite, în viitor, introducerea unei funcționalități suplimentare destinate tutorilor sau părinților, care să poată accesa în mod controlat și securizat caietele elevilor aflați în îngrijirea lor.

### **3.4.3. Comunicarea server - tableta: înrolarea**

Pentru a permite asocierea sigură a fiecărei tablete cu un cont de utilizator din platforma EVONOTE, a fost implementat un mecanism de înrolare (enrollment) în doi pași. Această abordare este necesară în contextul arhitecturii sistemului, unde serverul nu inițiază niciodată comunicarea — toate solicitările provin exclusiv dinspre dispozitiv.

Identificatorul duid (device unique ID), generat local la prima rulare a aplicației, este considerat garantat unic deoarece combină un șir aleatoriu de 20 de caractere cu timestamp-ul în milisecunde. Acesta este transmis serverului pentru inițierea procesului de înrolare.

Generarea codului de înrolare (6 caractere alfanumerice) și a cheii dkey (device key) este realizată strict pe server, tocmai pentru a garanta unicitatea și controlul procesului.

#### **Pe tabletă (la prima rulare):**

- Este inițializată baza de date locală și este creată o intrare care conține datele hardware ale dispozitivului.
- Se generează un identificator unic de înrolare duid, format prin concatenarea unui șir aleator de 20 de caractere cu timestamp-ul curent.
- Flag-ul enroll\_conf este setat pe 0, indicând că dispozitivul nu este încă înrolat.
- Aplicația verifică acest flag și, dacă înrolarea nu este confirmată, se lansează activitatea EnrollDevice.

**EnrollDevice:**

- Trimite un request către endpoint-ul /ws2024/enroll.php, conținând duid.
- Serverul generează un cod unic de înrolare enroll\_code (6 caractere alfanumerice majuscule) și cheia device-ului (dkey); codul de înrolare este valabil timp de 15 minute și îl afișează pe ecran.
- În baza acestui cod, pe server este reținută corespondența duid – dkey – enroll\_code, dar dispozitivul nu este încă activ.

**Pe platforma web:**

- Utilizatorul se autentifică și accesează secțiunea de administrare a dispozitivelor.
- Introduce codul afișat de tabletă și atribuie un nume identificabil (ex. „Tableta Andrei”).
- Dacă codul este valid și în termen, serverul finalizează asocierea și generează intrarea corespunzătoare pentru device, incluzând dkey, care va fi folosită ulterior pentru autentificare.

**Înapoi pe tabletă:**

- Utilizatorul apasă butonul de confirmare.
- Aplicația trimite un nou request pentru verificarea înrolării pe baza dkey și primește ca răspuns 0 sau 1
- Dacă înregistrarea există și este validă, flag-ul enroll\_conf este setat pe 1, iar aplicația continuă în mod normal.

### **3.4.4. Comunicarea server - tableta: sincronizarea caietelor / materiilor**

Sincronizarea listei de caiete se face manual din activitatea principală Dashboard, prin apăsarea butonului „Descarcă lista de caiete din cloud”. Aplicația trimite un request către endpoint-ul /ws2024/get-books.php, folosind dkey pentru autentificare. Serverul răspunde cu un obiect JSON conținând lista actualizată de caiete, care este apoi salvată local.

### **3.4.5. Comunicarea server - tableta: trimiterea scrierii către server**

Transmiterea datelor de scriere către server se realizează prin apelarea endpoint-ului /ws2024/incoming-writing.php. Rândurile sunt extrase individual din baza locală și

convertite într-un obiect JSON care conține dkey, skey (cheia unică a transmiției) și o listă de obiecte corespunzătoare rândurilor selectate.

Pentru fiecare rând sunt incluse următoarele câmpuri: ID (cheie primară locală), IDbook, page, listStop (indicator de întrerupere a liniei), type (1 = scriere, -1 = ștergere), timeWrite (timestamp) și listPts, o listă de puncte într-un format compact propriu care reține coordonatele x și y, presiunea aplicată și timpul de scriere în milisecunde.

Serverul validează dkey și skey, înregistrează datele și răspunde cu confirmare. Dacă răspunsul este valid, rândurile sunt marcate ca transmise. Sistemul include un mecanism de protecție care limitează la maximum trei încercări transmiterea unui rând în cazul apariției unor erori necunoscute.

### **3.4.6. Recompunerea scrierii**

Recompunerea paginilor scrise se realizează atât pe server (pentru generarea imaginilor de previzualizare și PDF), cât și pe tabletă (pentru afișare locală), utilizând un algoritm similar bazat pe ordinea cronologică a punctelor.

Pentru o anumită pagină, sunt selectate toate rândurile asociate din baza de date, în ordinea câmpului time\_write. Această ordine este esențială, deoarece asigură redarea fidelă a acțiunilor utilizatorului în succesiunea reală a evenimentelor (scriere sau ștergere) și permite acțiuni succesive diferite în aceeași zonă — de exemplu: scriere, ștergere, din nou scriere, apoi din nou ștergere, fără pierderea contextului.

Fiecare rând conține o listă de puncte care descriu traiectoria stylus-ului, folosite pentru a genera linii: între două puncte consecutive se parcurge traseul pas cu pas, în funcție de distanță, și se desenează puncte circulare (elipse) de grosime variabilă în funcție de presiune. Rezultatul este o linie continuă, realistă vizual, care redă fidel stilul de scriere.

Un câmp suplimentar indică dacă linia curentă este conectată cu cea următoare. Dacă nu există întrerupere, punctele sunt reunite într-o singură traiectorie. Câmpul tip determină dacă linia este o scriere (1) sau o ștergere (-1). În cazul ștergerii, algoritmul aplică o formă mai mare, de tip „gumele de șters”, peste zona afectată.

Pe server, imaginea finală este generată la o scară multiplă față de rezoluția normală, apoi redimensionată în jos. Această tehnică permite obținerea unei imagini mult mai netede (anti-aliasing), evitând aspectul pixelat specific desenului direct pe o rasterizare de bază.

După generare, imaginea este stocată și reutilizată ulterior, fără a fi recreată la fiecare acces. Regenerarea are loc doar dacă apar modificări noi în baza de date pentru pagina

respectivă, ceea ce permite economisirea resurselor serverului și reducerea timpilor de procesare.

Pentru generarea PDF-ului unui caiet, din motive de performanță și din cauza limitărilor PHP în gestionarea proceselor de lungă durată, este utilizată o strategie incrementală: fiecare pagină este procesată individual, prin apeluri AJAX către scripturi dedicate. Se verifică mai întâi dacă imaginea asociată paginii există și este actualizată; dacă nu, aceasta este (re)generată; la final un script assemblează PDF-ul într-o singură etapă.

## 3.5. Analiza resurselor consumate

Testele de consum au fost efectuate pe un dispozitiv BOOX Note 2 cu 4 GB RAM și 64 GB stocare internă, conform specificațiilor oficiale [15].

### 3.5.1. Analiza consumului de memorie heap

Pentru a evalua consumul de memorie al aplicației EVONOTE, am realizat o serie de teste de profilare heap utilizând instrumentele oferite de Android Studio Profiler. Scopul a fost identificarea consumului efectiv de memorie în Java Heap, în diferite scenarii de utilizare, precum și observarea comportamentului aplicației în raport cu creșterea numărului de pagini și a volumului de scriere.

Metrica centrală analizată a fost *Retained Size*, care reprezintă memoria totală ce ar fi eliberată dacă un anumit obiect activ și toate obiectele accesibile doar prin acesta ar fi colectate de Garbage Collector. Această valoare oferă o estimare precisă a cantității de memorie „reținută” de obiectele active din aplicație, excluzând memoria deja eliberată sau reutilizată.

Am urmărit comportamentul aplicației în trei scenarii distincte:

- Aplicație proaspăt instalată, o singură pagină, fără text: *Retained Size*  $\approx$  7.193 bytes (~7 KB)
- 10 pagini create, dintre care 2 scrise în proporție de ~50%: *Retained Size*  $\approx$  16.807 bytes (~16 KB)
- 20 pagini create, dintre care 3 scrise în proporție de ~50%: *Retained Size*  $\approx$  10.485 bytes (~10 KB)

La prima vedere, rezultatele pot părea contraintuitive — consumul pare să scadă odată cu creșterea numărului de pagini. Explicația constă în faptul că aplicația nu menține

simultan în memorie toate datele legate de paginile existente.

Acest comportament dovedește o gestionare eficientă și scalabilă a memoriei, în acord cu filosofia minimalistă a aplicației EVONOTE. Pe baza acestor teste, putem concluziona că aplicația folosește cel mult 20 KB de memorie reținută în heap-ul Java, chiar și în condiții de utilizare extinsă. Împreună cu memoria nativă aferentă SDK-ului și ecranului EPD (estimativ ~100 MB), consumul total de RAM rămâne în jurul valorii de 120–130 MB.

### **3.5.2. Analiza procesorului (CPU)**

În cadrul unui test de profilare CPU cu o durată de 27,5 secunde, am simulat utilizarea aplicației EVONOTE prin simpla navigare între pagini ce conțin un volum semnificativ de date (ex. paginile 3, 10, 20 și 21).

Analiza a arătat că:

- Firul principal al aplicației (eu.evonote.cloud) rămâne activ doar în rafale scurte, fără blocaje sau perioade prelungite de utilizare intensă a procesorului.
- Thread-ul de randare (RenderThread) funcționează într-un regim stabil, fără suprasarcini vizibile.
- Graficul global de utilizare CPU indică o curbă plată, fără spike-uri semnificative, ceea ce confirmă eficiența procesării la încărcarea paginilor deja existente.

Concluzia este că aplicația are un profil CPU foarte bun în utilizarea pasivă (navigare), ceea ce contribuie direct la economia de energie și la rularea fluentă pe dispozitive cu procesor modest.

Pentru a evalua consumul de procesor în timpul interacțiunii active, am realizat o sesiune de profilare CPU de 30 de secunde, în care s-a simulat scrierea rapidă pe o pagină (două rânduri de text, trasate continuu cu stylusul). Analiza a arătat că firul principal al aplicației (eu.evonote.cloud) devine activ la fiecare interacțiune, procesând rapid inputul și inițiind randarea în timp real, fără întârzieri vizibile.

Graficul global al utilizării CPU nu a prezentat spike-uri majore, menținându-se sub 25%.

### **3.5.3. Estimarea spațiului ocupat de datele de scriere**

Pentru a evalua spațiul de stocare necesar unui utilizator obișnuit, am efectuat o serie

de măsurători directe asupra bazei de date locale SQLite. Am analizat o pagină cu 6 rânduri din 17 posibile scrise integral. În acest caz, aplicația a generat:

- 690 de înregistrări în tabelul v\_writing;
- 1.5 MB spațiu ocupat în baza de date.

Extrapolând aceste date, o pagină complet scrisă produce în medie aproximativ 2000 de rânduri și ocupă ~4.5 MB. Pe baza acestei valori, putem estima consumul anual de spațiu pentru un elev obișnuit astfel:

- 10 materii principale, fiecare cu un caiet digital propriu;
- 50 de pagini complet scrise / materie / an;
- Rezultând în 500 de pagini  $\times$  4.5 MB = ~2.2 GB / an școlar.
- Aceleași 500 pagini = 1.000.000 intrări în baza de date

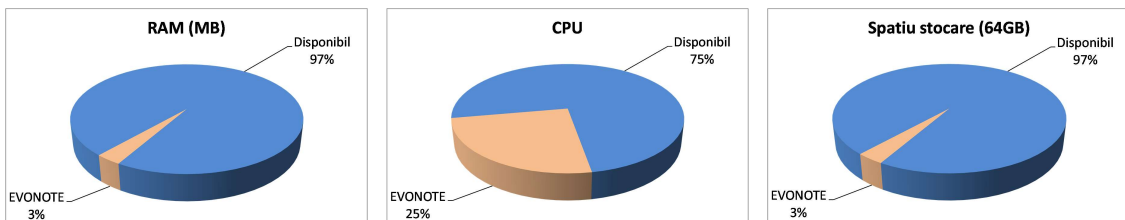
Această valoare include atât datele efective (lista de puncte scrise), cât și metadatele structurale generate de motorul SQLite.

Prin urmare, chiar și în absența oricărei arhivări sau curățări, stocarea necesară pentru un an întreg de utilizare rămâne modestă. Mai mult, un dispozitiv cu 64 GB de memorie internă poate acomoda fără dificultăți peste 20 de ani de școală, iar sincronizarea în cloud asigură redundanță și securitate suplimentară.

Merită menționat că motorul SQLite poate gestiona în mod stabil baze de date de ordinul gigabyților și tabele cu peste 2 miliarde de rânduri ( $2^{31}-1$ ). În practică, aplicații mobile cu baze de date de 2–4 GB funcționează fără probleme, cu condiția existenței unor indexuri eficiente.

În acest context, volumul generat de EVONOTE — estimat la ~2.2 GB per an — se încadrează confortabil în limitele suportate și nu ridică probleme de performanță sau stabilitate.

### 3.5.4. Concluzie: Impactul total asupra resurselor



**Figura 3.2:** Distribuția consumului de RAM, CPU și spațiu de stocare de către aplicația EVONOTE în timpul utilizării normale.

Graficul din stânga arată că EVONOTE consumă aproximativ 130 MB din cei 4096 MB disponibili, adică ~3.2% din memoria RAM.

Graficul central reflectă utilizarea procesorului în timpul scrierii: aplicația folosește în medie ~25% din capacitate, echivalentul a 2 nuclee din 8 activate temporar, fără suprasarcini sau blocaje.

Graficul din dreapta ilustrează spațiul necesar în stocarea internă: aproximativ 2.2 GB per an școlar, adică 3.4% din spațiul total de 64 GB. Acest volum permite păstrarea a peste 20 de ani de caiete digitale pe același dispozitiv.

## 4. Limitări, concluzii și direcții viitoare

Orice proiect realist trebuie să țină cont nu doar de obiectivele atinse, ci și de limitările sale — fie ele de natură tehnologică, economică sau practică. În cazul prototipului EVONOTE, pot fi identificate următoarele obstacole:

- Disponibilitatea restrânsă a hardware-ului EPD
- Prețul ridicat al dispozitivelor existente
- Lipsa de suport oficial din partea producătorilor
- Nevoia de standardizare a aplicației și a formatelor

Cu toate acestea, lucrarea de față a demonstrat că dezvoltarea unui caiet digital minimalist pe bază de tehnologie EPD este nu doar posibilă, ci și eficientă. Prototipul EVONOTE funcționează fluent pe hardware modest, implementează scrierea cu latență minimă și sincronizare în cloud, toate construite cu resurse limitate și logică simplă.

Dincolo de realizările tehnice, proiectul propune o viziune pragmatică asupra digitalizării educației — una centrată pe simplitate, accesibilitate și bunăstarea elevului. Limitările identificate pot fi depășite prin parteneriate, adaptări și — în perspectivă — dezvoltarea unui dispozitiv dedicat.

În plus, pot fi imaginate extensii utile care păstrează filosofia de bază:

- Conectivitate mobilă prin SIM
- Modul GPS pentru siguranța copiilor
- Acces controlat pentru tutori sau profesori

Toate aceste direcții susțin ideea unui instrument educațional minimalist, dedicat, stabil și scalabil.

# Bibliografie

- [1] MIT News, *In the Blink of an Eye: The Limits of Latency Perception*, 16 ianuarie 2014, <https://news.mit.edu/2014/in-the-blink-of-an-eye-0116>
- [2] Microsoft Applied Sciences Group, *High Performance Touch* [video], martie 2012, <https://www.youtube.com/watch?v=vOvQCPLkPt4>
- [3] ALLSTARSPACE, Samsung Galaxy Tab S6 Lite Review – Writing and Stylus Lag Demo [video], YouTube, secțiunea relevantă de la minutul 9:58, [https://youtu.be/Mea9ezy9\\_zs?si=WXcHk-Vh3UMphReI&t=598](https://youtu.be/Mea9ezy9_zs?si=WXcHk-Vh3UMphReI&t=598)
- [4] Reddit, *Could anyone post Galaxy S6 Lite S-Pen latency?*, postare din 17 iulie 2020 pe r/GalaxyTab, care conține un videoclip test de latență (filmare la 240 fps) găzduit pe Google Drive, [https://www.reddit.com/r/GalaxyTab/comments/hsws4a/could\\_anyone\\_post\\_galaxy\\_s6\\_lite\\_spen\\_latency/](https://www.reddit.com/r/GalaxyTab/comments/hsws4a/could_anyone_post_galaxy_s6_lite_spen_latency/)
- [5] Parkablogs, *Artist review: Lenovo P11 Pro with Precision Pen 2*, 13 septembrie 2021, <https://www.parkablogs.com/content/artist-review-lenovo-p11-pro-precision-pen-2>
- [6] Apple Discussions, *Latency problem with Apple Pencil on iPad Pro*, <https://discussions.apple.com/thread/255642920?sortBy=rank>
- [7] reMarkable, *About reMarkable 2*, articol de suport oficial, <https://support.remarkable.com/s/article/About-reMarkable-2>
- [8] E Ink, *How E Ink works*, pagină oficială de prezentare tehnologică, [https://www.eink.com/tech/detail/How\\_it\\_works](https://www.eink.com/tech/detail/How_it_works)
- [9] Android Developers, *Stylus Input*, <https://developer.android.com/develop/ui/views/touch-and-input/stylus-input>
- [10] Saumay Pushp et al., *Characterizing and Adapting User-Perceived Latency of Android Apps*, <https://saumaypushp.github.io/files/ShuffleDog.pdf>

- [11] Android Open Source Project, *Implement VSYNC*, <https://source.android.com/docs/core/graphics/implement-vsync>
- [12] Rabbani, S. A., Ahmed, M., Zahid, A. *E-Ink: Revolution of Displays*. MATEC Web of Conferences 381, 02003 (2023), <https://doi.org/10.1051/mateconf/202338102003>
- [13] Onyx International, *Onyx Android Demo – demonstrații oficiale pentru SDK-ul BOOX*, <https://github.com/onyx-intl/OnyxAndroidDemo>
- [14] Business Wire, *Harvard Study Shows E Ink's ePaper Is Up to Three Times Healthier for Your Eyes Than LCD Screens*, publicat la 13 martie 2023 <https://www.businesswire.com/news/home/20230313005152/en/Harvard-Study-Shows-E-Inks-ePaper-Is-Up-to-Three-Times-Healthier-for-Your-Eyes-Than-LCD-Screens>
- [15] Onyx Boox, *BOOX Note2 Specifications*, [https://onyxboox.com/boox\\_note2](https://onyxboox.com/boox_note2)
- [16] Datchuk, S. (2024). *Handwriting: Beneficial to Reading and Often Misunderstood*. Iowa Reading Research Center, <https://irrc.education.uiowa.edu/blog/2024/07/handwriting-beneficial-reading-and-often-misunderstood>
- [17] E Ink. (2024). *E Ink in Education: The Rise of Paperless, Glare-Free Learning*, <https://blog.eink.com/e-ink-in-education>

NOTĂ: toate link-urile au fost accesate și verificate în data de 7 iunie 2025.

# Anexe

## Anexa 1 – Callback-ul `getRawInputCallback()`

Aplicația EVONOTE folosește SDK-ul Onyx pentru a activa modul `RawDrawing`, care permite scrierea directă pe ecranul EPD cu latență minimă. Pentru a intercepta datele scrise cu stylus-ul, se implementează metoda `getRawInputCallback()`, care returnează un obiect `RawInputCallback`.

Aceasta oferă următoarele metode principale:

- `onBeginRawDrawing(...)` – începutul unei linii;
- `onRawDrawingTouchPointMoveReceived(...)` – puncte în mișcare;
- `onRawDrawingTouchPointListReceived(...)` – listă completă de puncte la finalul unei linii;
- `onEndRawDrawing(...)` – finalul gestului de scriere;
- `onBeginRawErasing(...)` și `onEndRawErasing(...)` – începutul și sfârșitul ștergerii;
- `onRawErasingTouchPointMoveReceived(...)` și
- `onRawErasingTouchPointListReceived(...)` – punctele traseului de ștergere.

Fiecare punct include coordonate (x, y), presiune, dimensiune și timestamp, toate fiind salvate local pentru sincronizare ulterioară.

Această structură este documentată în proiectul oficial Onyx Android Demo [13].

```
public RawInputCallback getRawInputCallback() {
    if (rawInputCallback == null) {
        rawInputCallback = new RawInputCallback() {
            @Override
            public void onBeginRawDrawing(boolean b, TouchPoint touchPoint) {
                // begin of stylus data
                pointsDrawSave.clear();
                String pointDet = touchPoint.getX()+"|"+
                    touchPoint.getY()+"|"+
                    touchPoint.getPressure();
                pointsDrawSave.add(pointDet);
            }
        }
    }
}
```

```

@Override
public void onRawDrawingTouchPointMoveReceived(TouchPoint touchPoint) {
    // stylus data during stylus moving
    String pointDet = touchPoint.getX()+"|"+
        touchPoint.getY()+"|"+
        touchPoint.getPressure();
    pointsDrawSave.add(pointDet);
    if (pointsDrawSave.size()>=30) {
        savePoints(String.valueOf(pointsDrawSave), 1,0);
    }
}

@Override
public void onRawDrawingTouchPointListReceived(TouchPointList touchPointList) {
    // full list of points after lifting the stylus (end of stroke)
    savePoints(String.valueOf(pointsDrawSave), 1,1);
    TouchPoint point;
    List<TouchPoint> points = new ArrayList<>();
    for (int i = 0; i < touchPointList.size(); i++) {
        point = touchPointList.get(i);
        points.add(point);
    }
    writeBitmap(points);
}

@Override
public void onEndRawDrawing(boolean b, TouchPoint touchPoint) {
    // end of stylus data
}

// Erasing-related callbacks have been omitted from this annex.
// Only the main RawDrawing methods are shown here.

};
}
return rawInputCallback;
}

```

## Anexa 2 – Secvențe de cod semnificative

### Redare grafică pe canvas-ul intermediar din memorie

```
private void writeBitmap(List<TouchPoint> points) {
    if (canvas == null || points == null || points.isEmpty()) return;
    boolean isFirstPoint = true;
    float prevX = 0, prevY = 0;
    for (TouchPoint point : points) {
        float x = point.getX();
        float y = point.getY();
        float pressure = point.getPressure();
        float dynamicStrokeWidth = (float)Math.ceil(pressure * 0.02f) / 10f;
        renderPaint.setStrokeWidth(dynamicStrokeWidth);
        if (isFirstPoint) {
            isFirstPoint = false;
        } else {
            canvas.drawLine(prevX, prevY, x, y, renderPaint);
        }
        prevX = x;
        prevY = y;
    }
}
```

### Redare grafică pe suprafața de afișare a ecranului

```
private void drawBitmap() {
    if (surfaceView.getHolder() == null) {
        return;
    }
    Canvas canvas = surfaceView.getHolder().lockCanvas();
    if (canvas == null) {
        return;
    }
    EpdController.enablePost(surfaceView, 1);
    Rect rect = new Rect(0, 0, surfaceView.getWidth(), surfaceView.getHeight());
    canvas.drawRect(rect, clearPaint);
    canvas.drawBitmap(bkGroundBitmap, null, rect, renderPaint);
    canvas.drawBitmap(renderBitmap, 0, 0, renderPaint);
    surfaceView.getHolder().unlockCanvasAndPost(canvas);
}
```

## Anexa 3 – Imagini și capturi de ecran



Figura A.1: Dispozitivul EPD utilizat.

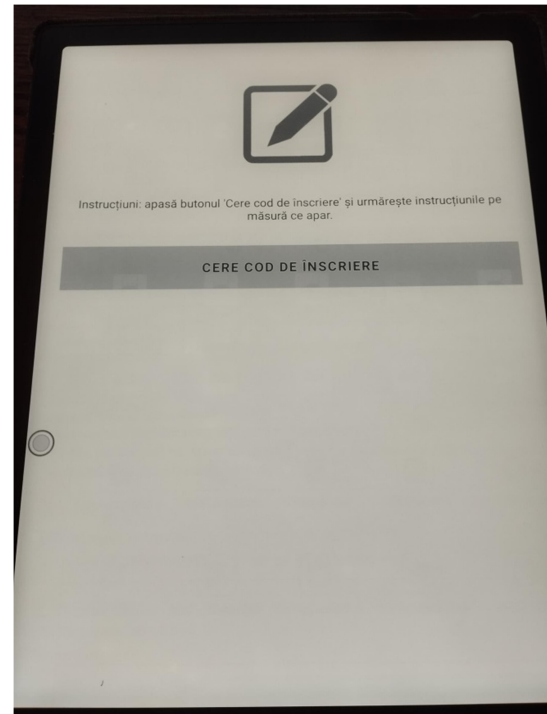


Figura A.2: Pasul 1 de înrolare.

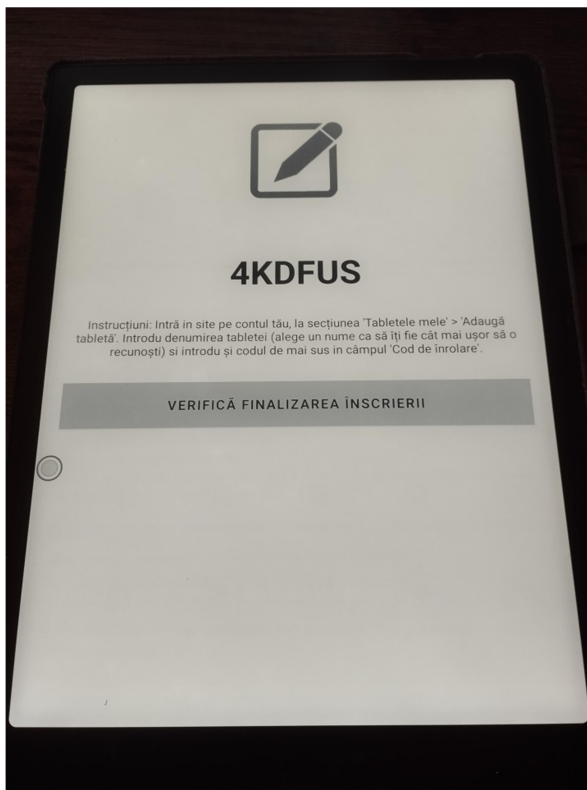


Figura A.3: Pasul 2 de înrolare.



Figura A.4: Pasul 3 de înrolare, pe server.

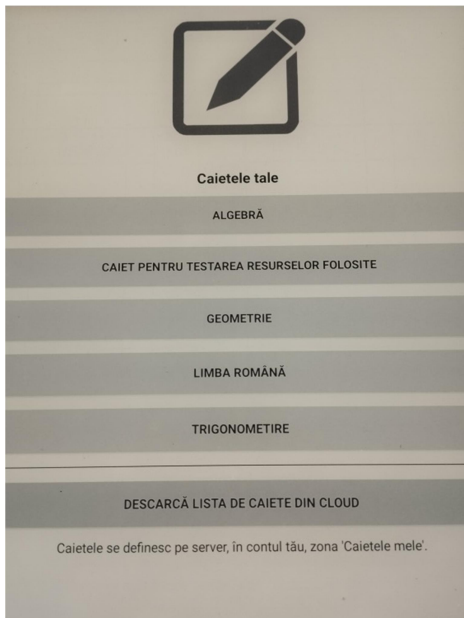


Figura A.5: Dashboard – lista caietelor.



Figura A.6: O pagina scrisă – a se observa meniul minimalist, doar cu butoanele esențiale.

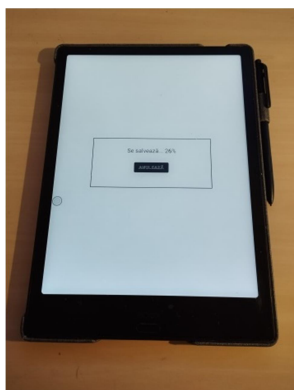


Figura A.7: Salvarea datelor în cloud

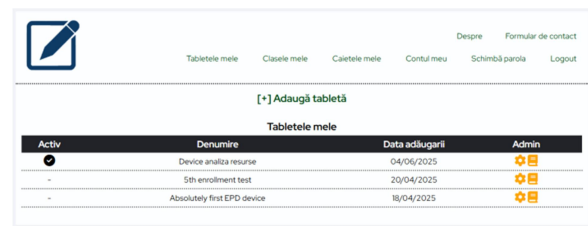


Figura A.8: Lista device-urilor utilizatorului.

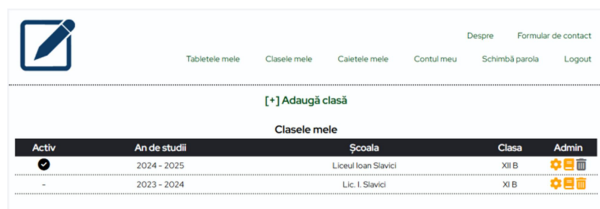


Figura A.9: Lista claselor utilizatorului

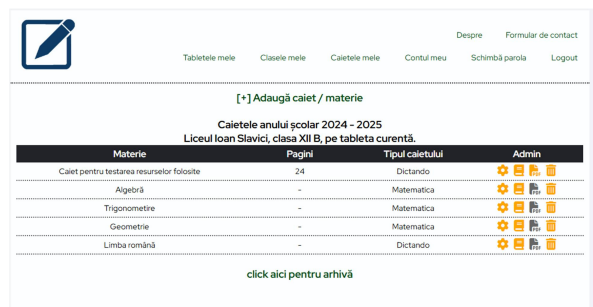


Figura A.10: Lista caietelor – clasa și anul curent.

Caiet pentru testarea resurselor folosite  
Anul școlar 2024 - 2025, Liceul Ioan Slavici, clasa XII B

« pagina anterioară

Pagina 24 din 25

Mergi la pagina: 24 ▾

pagina următoare »

Magia unei zile de primăvară

Departate de agitația orașelor, sate  
pitouști se trezesc la viață.  
Verdele 'cud al frunzisului, înflo-  
-rirea copacilor și a florilor, păsă-  
-rite care cîmpesc vesel, toate ne  
încorjoară cu o energie proaspătă  
și revigorantă. Este o renaștere a  
vieții după anotimpul rece.

Figura A.11: Pagina scrisă din figura A6 văzută în interfața web