

MemorEasy

Contents

Ideea proiectului (proiect).....	1
Arhitectura proiectului	3
Aspecte tehnice interesante si/sau dificile.....	4
Ce mai avem de implementat.....	5

Ideea proiectului (backlog)

Proiectul are la baza simpla idee a unei aplicatii de tip flashcard, pe care o extinde printr-un design interactiv, o viitoare capacitate gratuita de a adauga imagini (ceea ce nu regasim la alternativele existente pe piata), si prin avantajul de a fi utilizata concurent de mai multi useri (un numar aproximativ nelimitat, fara a amesteca informatiile fiecaruia) pe acelasi dispozitiv, dar si prin viitoarele mici teste de memorie generate pornind de la flashcard-urile existente.



Etapele crearii proiectului sunt intr-o ordine mai mult sau mai putin exacta, oferirea:

- unui mediu de logare pentru securizarea datelor
- capacitati de a crea, sterge si edita categorii, capitole si flashcard-uri pentru o buna organizare a datelor
- capacitati aplicatiei de a functiona in intregime off-line datorita dependintelor din mediul de lucru
- capacitati de a depozita intr-o forma utila datele intr-o baza de date asociata prin dependinte si de a le refolosi cand este necesar
- optiunii de a seta modul Light / Dark, pentru a asigura un confort al ochilor in functie de timpul zilei
- optiunii de a asocia imagini flashcard-urilor, pentru un suport vizual adaugat celui scris
- optiunii de a testa cunostintele prin teste automate
- optiunii de a importa flashcard-urile in diferite formate (.csv etc.)
- (poate) unui depozit 'remote' in cloud.

Arhitectura proiectului

Am ales sa folosim Java in mediul oferit de IDE-ul IntelliJ. Partea grafica si interactiva a fost creata cu ajutorul tehnologiei JavaFX in combinatie cu SceneBuilder, iar persistenta datelor a fost asigurata de SGBD-ul MySQL.



Logo Scene Builder

Am optat pentru Java datorita particularitatilor OOP ale sale, necesare dezvoltarii unei aplicatii cu entitati care interactioneaza intre ele, dar si datorita compatibilitatii cu JavaFX, cu SGBD-ul MySQL, dar si accesibilitatii limbajului. In ciuda faptului ca JavaFX e o tehnologie destul de putin folosita, am ales-o pentru ca este usor de invatat si de folosit, mai ales

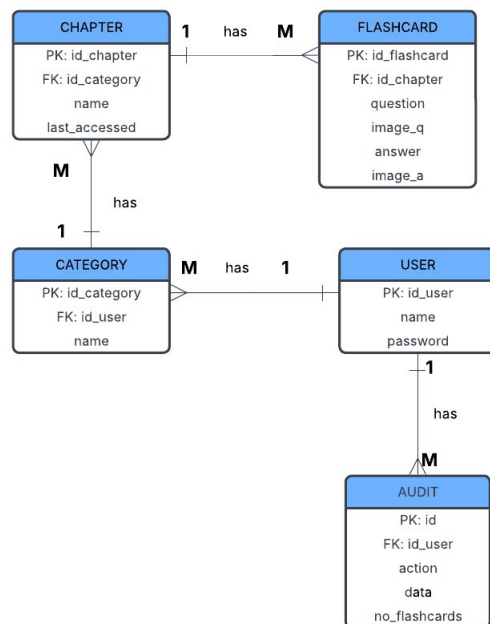
pentru
u ca
vine
„la
pache
t” cu
uneal
ta



Scene Builder, capabila sa genereze cod in fisierele *.xml* asociate controller-elor de tip *.java*, in timp real, cand programatorul plaseaza cu drag&drop obiecte si containere in scena.

Ca atare, arhitectura proiectului este constituita din mai multe pachete si directoare descriptive (Controllers, Models, Utils, Images, Styles, Views), care contin fisiere *.java* (controller-e si modele), *.xml* (view-uri), *.css* si imagini.

Modelele sunt Category, Chapter, Flashcard, User si (in viitor) Audit, fiecare avand controller-e asociate pentru scenele redate in view-uri.



Logo JavaFX

Aspecte tehnice interesante si/sau dificile

Printre aspectele tehnice greu de gestionat s-au remarcat:

- testarea aplicatiei: fiecare coechipier are propria baza de date locala, asa ca testarea aplicatiei in toate cazurile posibile a devenit o provocare din moment ce datele nu erau omogene
- imparirea sarcinilor: a devenit si aceasta dificila, intrucat multe functii si optiuni sunt interdependente
- gestionarea workflow-ului: rezolvarea conflictelor si gestionarea branch-urilor
- invatarea intr-un timp scurt a unor tehnologii noi.

Printre aspectele tehnice interesante s-au remarcat:

- combinarea tehnologiilor JavaFX si SceneBuilder care au permis crearea rapida a unei interfete intuitive.
- asigurarea persistentei datelor intr-un mediu sigur si organizat folosind MySQL
- folosirea mostenirii, encapsularii si polimorfismului in Java pentru o structura modulara
- adaugarea in *pom.xml* a dependintelor necesare proiectului
- crearea animatiei de flashcard
- implementarea interfetei *Initializable* care incarca elementele scenei pentru continut dinamic (exemplu: butonul care redirectioneaza catre capitole, indicat de pagination).

Ce mai avem de implementat

Mai avem de implementat 7 user stories din 20 initiale:

- in ceea ce priveste flashcard management:
 - adaugarea de imagini pentru flashcard-uri.
- in ceea ce priveste testele:
 - crearea de teste automate de tip matching;
 - crearea de teste automate de tip „fill the missing words”;
 - revizuirea tuturor flashcard-urilor intr-o ordine aleatoare;
 - repetitia spatiata.
- in ceea ce priveste importul si exportul datelor:
 - crearea de fisiere .csv pentru backup-ul datelor, precum si importarea de fisiere .csv pentru a crea noi date (flashcard-uri);
 - capacitatea de a stoca flashcard-urile in cloud (posibil).