

Software Architecture Report

JANUARY 27

App Version 2.0
Commit 4d82ed2



Logo
Name

Scopul proiectului

Proiectul este o aplicatie educationala pentru elevii/studentii interesati de o facultate sau cariera in IT. Programul dispune de lectii, probleme si informatii generale despre algoritmica, structuri de date si alte cunostinte necesare.

Momentan, programul contine:

- meniul pentru selectarea capitolului/lectiei
- lectii despre diferite concepte de computer science, in principal algoritmica si structuri de date
- teste de antrenament (quizuri)
- probleme de diferite dificultati la fiecare capitol (cu rezolvare)
- cornometrare pentru teste
- rubrici pentru informatii suplimentare (admitere, cariere, linkuri utile)

Pe viitor, vom implementa in program:

- to be determined

Ghiduri necesare unui nou developer

Rularea locala+build:

Fisierul "cod dupa al n-lea sprint.cpp" contine tot codul aplicatiei, trebuie doar descarcat local de pe git. Pentru build+rulare, este nevoie de un compiler de c++ (preferabil cel din colectia gnu) si rulata in terminal comenzile:

```
g++ numefisier.cpp -o numeexecutabilcreat.exe  
./numeexecutabilcreat.exe
```

Sau, intr-un IDE ca si CodeBlocks, daca e configurat compilerul se poate folosi pur si simplu butonul de build and run.

Deploy pe alte masini

Executabilul creat dupa compilare este suficient pentru a rula programul.

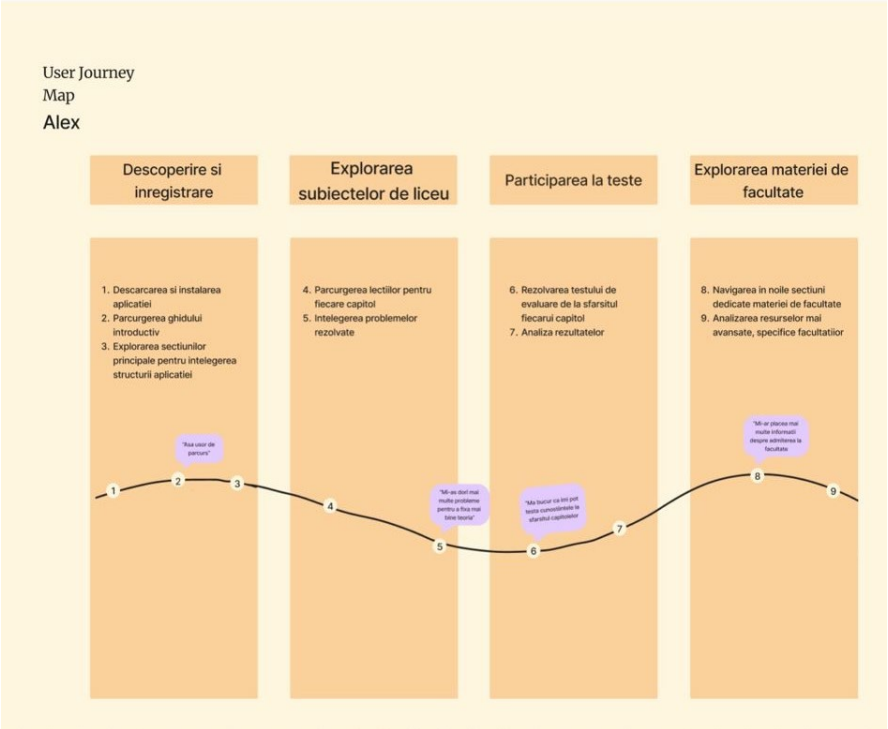
Cum contribuim de obicei la aplicatie

Destul de straightforward, indentare si coding standards ca si restul codului (indentare 4 spatii, paranteze etc), iar pentru rubricile cu informatii trebuie tinut cont de numarul de spatii care variaza (se observa din print-uri).

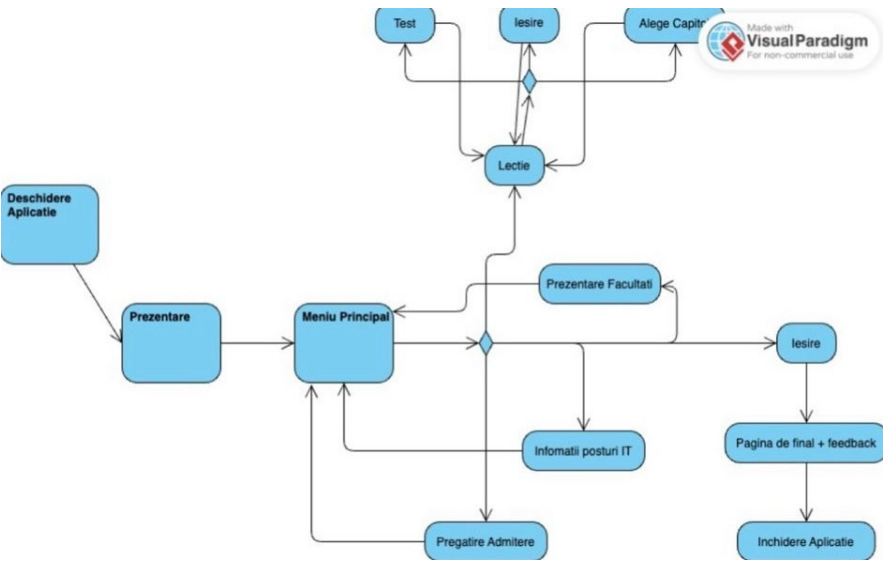
Application entry points

Datele de intrare in aplicatie sunt exclusiv din meniuri switch/case, numere sau litere date de la tastatura pentru a selecta optiunile din meniu/variantele de raspuns la quizuri.

Diagrame High-Level ale arhitecturii



User journey map



Activity diagram

Deployment plan

Aplicatia este facuta pentru a fi deployed local, dupa descarcarea executabilului si rulat in consola.

Procesul de testare

Testele verifica daca toate optiunile din meniuri functioneaza si redirectioneaza userul unde trebuie si daca functioneaza cronometrul pentru quizuri.

Nu am stabilit un CI pipeline pentru ca aplicatia este destul de simpla pentru teste clasice si deployment manual.

External dependencies

Folosim librariile urmatoare:

`<iostream>`

– fluxuri de intrare/iesire

`<iomanip>`

`<chrono>`

– pentru cronometru

`<windows.h>`

`<mmsystem.h>`

– manipulare api-uri windows, in principal pentru `system("pause")` si `system("cls")`

`<string.h>`

– manipulare string-uri

`<queue>`

– structuri de date folosite in exemplele din lectii

Concluzii

Arhitectura prezinta o baza pentru un viitor program educational. E simplu dar vrea sa faca invatarea eficienta.