Advanced Graph Visualizer

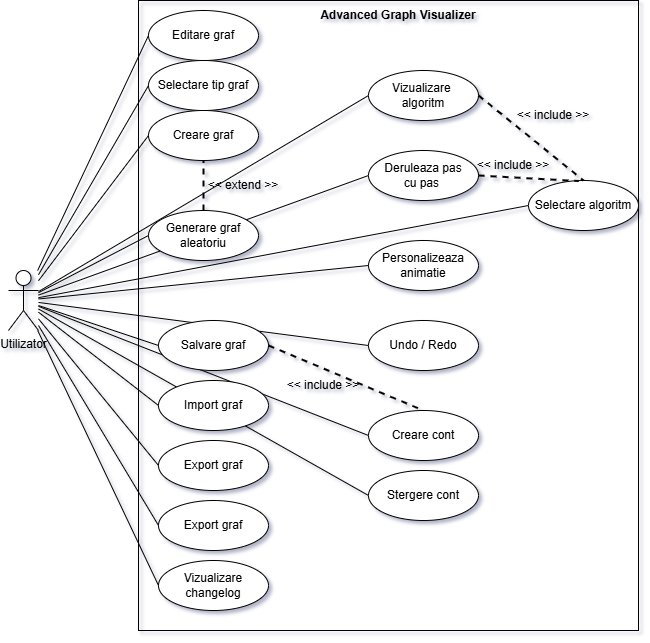
1. Se pornește de la un scenariu propriu care descrie cerințele sistemului ce urmează a fi dezvoltat (Requirements Document). Acesta va fi însoțit de o Diagramă de Cazuri de Utilizare (Use Case Diagram) [1,25,26]. (Responsabilitate: Product Owner)

Aplicația Advanced Graph Visualizer este o platformă interactivă care permite utilizatorilor să creeze, editeze și să vizualizeze grafuri într-un mod ușor și intuitiv. Utilizatorul poate construi grafuri fie dintr-un editor de text (de exemplu, prin introducerea unei matrici sau liste de adiacență), fie folosind o interfață grafică cu butoane pentru adăugarea și ștergerea de noduri sau muchii.

Sistemul oferă posibilitatea de a salva grafurile create, de a face import sau export de fișiere, dar și de a urmări modificările pas cu pas prin funcțiile undo și redo. Fiecare utilizator își poate crea un cont personal pentru a păstra configurațiile salvate și istoricul modificărilor.

Aplicația permite și vizualizarea animată a algoritmilor pe grafuri (precum Dijkstra, DFS, BFS etc.), cu explicații clare și o descriere educativă pentru fiecare algoritm. Utilizatorul poate personaliza animațiile, derula pașii manual și genera grafuri aleatorii pe baza unor parametri setați de el.

Scopul aplicației este de a combina partea educativă cu cea practică, oferind o modalitate modernă și atractivă de a învăța și înțelege conceptele de teorie a grafurilor.



2. Din Requirements Document vor fi derivate User Stories inițiale [1-5]. (Notă: cerințele sistemului vor fi alese astfel încât să rezulte cel puţin 15 User Stories.) (Responsabilitate: Product Owner)

1. Vreau să pot crea un graf și să îl pot edita, vizualiza și salva în contul meu, ca să nu-l pierd când închid aplicația.
2. Vreau să pot alege ce tip de graf creez (orientat, neorientat, ponderat) și să pot aplica algoritmi pe el imediat.
3. Vreau să pot vizualiza cum funcționează diferiți algoritmi (BFS, DFS, Dijkstra) și să pot modifica animațiile acestora.
4. Vreau ca fiecare algoritm să aibă explicații și descrieri, dar și o opțiune de derulare pas cu pas, pentru a învăța mai ușor.
5. Vreau să pot genera grafuri aleatorii și apoi să le modific manual, să le salvez și să le export dacă îmi plac.
6. Vreau să pot încărca grafuri din fișiere text, să le transform în format vizual și să le rulez algoritmi peste ele.
7. Vreau ca toate modificările să fie vizibile instant și să am un changelog complet cu undo, redo și link către fiecare versiune.
8. Vreau să pot interpreta un graf atât ca matrice, cât și ca listă de adiacență, și să pot comuta între aceste moduri în timpul vizualizării.
9. Vreau ca animațiile să fie personalizabile, dar și să existe o legendă care se schimbă în funcție de algoritm.
10. Vreau să am un cont în care se salvează toate grafurile.

3. User Stories de la punctul anterior vor fi rafinate astfel încât să satisfacă cele 6 criterii: Independent, Negotiable, Valuable to users or customers, Estimable, Small, Testable (INVEST) [1-3]. Să se justifice modificările făcute. Fiecărei User Story i se va atribui o prioritate, care va fi justificată. User Stories rezultate vor alcătui Project Backlog. (Responsabilitate: Product Owner și Dezvoltatorii).

1. Independent – sa nu depinda de alte user stories
2. Negotiable – detalii ajustabile intre PO si DEV
3. Valuable – valoros, sa aduca valoare reala produsului
4. Estimable – sa poata fi estimate de catre echipa
5. Small – sa poata fi implementat in 1-2 zile
6. Testable – sa existe o modalitate clara de verificare daca e gata

BACKLOG:

1. **Ca utilizator, vreau să pot crea un graf nou și să îl vizualizez pe ecran,** astfel încât să pot începe construcția structurii mele de date.
2. **Ca utilizator, vreau să pot alege tipul de graf (orientat, neorientat, ponderat etc.),** astfel încât să pot modela corect problema mea.
3. **Ca utilizator, vreau să pot crea un graf folosind un editor de text (matrice sau listă de adiacență),** astfel încât să pot lucra rapid dacă știu structura.
4. **Ca utilizator, vreau să pot crea un graf din interfața grafică (GUI),** astfel încât să pot adăuga sau șterge noduri și muchii cu ușurință.
5. **Ca utilizator, vreau să pot genera un graf aleatoriu cu parametrii setați de mine,** astfel încât să pot testa algoritmi pe diverse structuri.
6. **Ca utilizator, vreau să pot modifica graful (muta noduri, adăuga/șterge muchii),** astfel încât să pot ajusta structura vizuală.
7. **Ca utilizator, vreau să pot face UNDO și REDO,** astfel încât să pot reveni rapid la o stare anterioară a grafului.
8. **Ca utilizator, vreau să văd în timp real modificările făcute asupra grafului,** astfel încât să observ instant rezultatele acțiunilor mele.
9. **Ca utilizator, vreau să pot importa un graf dintr-un fișier .txt,** astfel încât să pot încărca grafuri existente.
10. **Ca utilizator, vreau să pot exporta un graf creat sau modificat,** astfel încât să îl pot salva local sau partaja.
11. **Ca utilizator, vreau să pot salva configurațiile de grafuri în contul meu,** astfel încât să le pot accesa ulterior.
12. **Ca utilizator, vreau să pot vizualiza un changelog al modificărilor,** astfel încât să pot urmări evoluția fiecărui graf.
13. **Ca utilizator, vreau să pot selecta un algoritm dintr-o listă (Dijkstra, BFS, DFS etc.),** astfel încât să pot învăța cum funcționează.
14. **Ca utilizator, vreau să pot vizualiza pas cu pas execuția algoritmului,** astfel încât să pot urmări cum se modifică graful în timp.
15. **Ca utilizator, vreau ca fiecare algoritm să aibă o descriere explicativă,** astfel încât să înțeleg logica lui.
16. **Ca utilizator, vreau să pot personaliza animațiile algoritmilor (culori, viteză, stil),** astfel încât să le pot adapta preferințelor mele.
17. **Ca utilizator, vreau să pot crea un cont,** astfel încât să pot salva și gestiona propriile grafuri și configurații personale.

4. Să se estimeze efortul (în story points) pentru fiecare User Story folosind Planning Poker [7,8] și să se justifice estimarea. Se va demonstra utilizarea Planning Poker. (Responsabilitate: Dezvoltatori)

---------

5. Se completeaza Sprint Backlog in Planning Meeting (care are loc in prima zi a Spint-ului). Observatie: În mod normal, Sprint Backlog nu va cuprinde toate User Stories din Project Backlog; din contră, având în vedere durata redusă a sprint-ului și dimensiunea echipei, este de așteptat ca doar o parte din ele să poată fi incluse în Sprint Backlog, selecția fiind făcută în funcție de estimarea efortului și priorități. (Responsabilitate: Dezvoltatori, Scrum Master, Product Owner).

---------

6. Se va folosi un tool (la alegerea echipei) pentru project management și issue tracking [5,12]. La prezentare, se va dovedi ca s-a utilizat tool-ul respectiv. (Responsabilitate: Dezvoltatori, Scrum Master, Product Owner)

< screenshot uri JIRA>

7. Scrum Master-ul, cu ajutorul dezvoltatorilor, va realiza urmatoarele diagrame (aceste diagrame vor fi incluse în Project Backlog):

a. Diagrama de workflow a proiectului.

b. Diagrama bazei de date (dacă este cazul)

A white paper with black text

AI-generated content may be incorrect.

A diagram of a computer

AI-generated content may be incorrect.

8. Product Owner-ul, cu ajutorul întregii echipe va dezvolta urmatoarele diagrame (aceste diagrame vor fi incluse în Project Backlog):

a. Diagrama UML de cazuri de utilizare.

b. Diagrama Gantt