

# Raport

Am început proiectul prin definirea clară a obiectivelor și a funcționalităților cheie. Am folosit platforma Trello pentru a crea o hartă vizuală a proiectului, cu liste și card-uri reprezentând diferitele etape și task-uri. Fiecare membru al echipei a avut acces la Trello, unde s-au asignat și actualizat task-urile în timp real. În plus, am dezvoltat o diagramă a bazei de date pentru a asigura o structură coerentă și eficientă a informațiilor. Împărțirea task-urilor s-a realizat având în vedere preferințele și dorințele fiecărui membru al echipei în privința ariei în care dorea să contribuie. Prin intermediul discuțiilor deschise, am identificat interesele și abilitățile individuale ale fiecărui membru și am împărțit task-urile în mod corespunzător, oferind astfel posibilitatea fiecărui să lucreze la componenta aleasă. Această abordare a condus la o distribuție echilibrată a sarcinilor, astfel încât fiecare membru să aibă responsabilități atât pe partea de backend, cât și pe cea de frontend.

GitHub a fost integrat pentru a facilita gestionarea codului sursă și colaborarea. Am creat un repository comun, unde fiecare membru din echipă a primit acces. Această abordare ne-a oferit o vedere detaliată asupra modificărilor și a permis fiecărui membru al echipei să aducă contribuții semnificative. Prin utilizarea acestei platforme, am reușit să menținem constant sincronizat codul sursă între membrii echipei, contribuind astfel la o colaborare fluidă și la o integrare eficientă a aporturilor individuale în proiect.

Link Github: <https://github.com/Razvan48/Project-Dezvoltarea-Aplicatiilor-Web-DAW>

Comunicarea a fost ușurată prin intermediul platformei Github, care dispune de un sistem prin care putem lăsa comentarii direct la liniile de cod. Fiecare membru a efectuat procese de revizuire a codului pentru modificările aduse de alții membri. Folosind sistemul menționat anterior, am reușit să detectăm rapid eventuale bug-uri apărute în implementările noastre.

Dezvoltarea a fost progresivă, cu funcționalități cheie adăugate în fiecare sprint. Am implementat un cadru de lucru bazat pe sprint-uri de 3 săptămâni, începând din momentul în care a fost postat documentul ghid "Progres proiect - Etapa I" pentru coordonarea proiectului de laborator.

## **SPRINT 1 (7 noiembrie - 28 noiembrie):**

- Categorii + Discuții + Răspunsuri + Comentarii + Roluri utilizatori

## **SPRINT 2 (28 noiembrie - 19 decembrie):**

- Sistem de votare + Profil utilizator + Summernote Editor

## **SPRINT 3 (19 decembrie - 9 ianuarie):**

- Sistem de notificări + Paginare + Sortare + Motor de căutare + Compilator + Sistem de cooldown + Gestionarea utilizatorilor

## **Provocări**

În cadrul proiectului am avut componente care depindeau de alte componente. Prin urmare, am întâmpinat dificultăți atunci când un membru al echipei a trebuit să aștepte după o

componentă implementată de altă persoană din echipă. Aceasta a creat uneori blocaje temporare și a întârziat progresul în unele etape ale proiectului.

O altă provocare cu care ne-am confruntat a fost dată de momentele în care mai mulți membrii lucrau simultan la proiect și încercau să încarce modificările lor pe Github prin operațiunea de commit. Acest lucru creează situații în care unii membrii din echipă nu aveau cea mai recentă actualizare în propriul lor cod. Astfel, persoana care nu avea ultimele modificări în codul său trebuia să efectueze un proces de merge între propriile sale modificări și ultima versiune a codului, pentru a evita pierderea sau conflictul de informații.

Pentru a gestiona problema întârzierii implementării componentelor, am adoptat următoarele abordări eficiente:

1. Am fragmentat codul în componente mai mici, care pot fi implementate individual și nu au un impact semnificativ asupra celorlalte. Acest lucru ne-a permis să avansăm treptat și să finalizăm implementările complete ale cerințelor proiectului.
2. În situațiile în care prima variantă nu a funcționat, am optat pentru colaborare activă. Chiar și având orare diferite, am reușit să găsim timp pentru discuții și luarea deciziilor în mod comun. În timpul vacanței, am continuat să colaborăm pentru a rezolva problemele și a ne încadra în deadline-uri.
3. Am adoptat o abordare proactivă prin notarea fiecărei probleme sau bug și prin încărcarea modificărilor în repository. Acest lucru permite altor membri ai echipei să identifice rapid problemele și să contribuie la rezolvarea lor.

Colaborarea eficientă și comunicarea transparentă au fost cheia pentru a gestiona în mod eficient întârzierile și pentru a ne asigura că proiectul avansează într-un mod coordonat.

În cadrul modificărilor simultane pe Github, a fost nevoie doar de o mai bună coordonare. Fiecare membru a spus când ceva e gata, când ceva poate lua mai mult timp sau când ceva nu merge și are nevoie de ajutor. Astfel am reușit să diminuăm părțile de conflict care apăreau la început. Am implementat o mai bună coordonare a task-urilor și am stabilit deadline-uri clare pentru fiecare componentă, asigurându-ne că colaborarea între membrii echipei este mai sincronizată și eficientă. Am încercat să evităm să avem commit-uri foarte mari care să modifice multe componente ale proiectului. În situațiile în care trebuiau editate mai multe componente, comunicam pentru a putea avea o singură persoană care lucrează, pentru ca apoi să fie preluată munca de alt membru, menținând codul sursa sincronizat.

## Productivitate

- 1. Realizările obținute** - Am integrat în proiectul nostru toate cerințele, împreună cu cerințele suplimentare.
- 2. Respectarea deadline-urilor interne** - Echipa noastră a gestionat eficient termenele limită stabilite, asigurându-ne că fiecare etapă a proiectului a fost finalizată la timp. Respectarea cu strictețe a cerințelor fiecărui sprint a contribuit la o dezvoltare eficientă și la atingerea obiectivelor stabilite într-un mod corespunzător.
- 3. Gestionarea și rezolvarea rapidă a bug-urilor** - Atunci când un bug era identificat, echipa a acționat prompt pentru a rezolva problema, asigurându-se că ceilalți membri pot continua munca fără a întâmpina constant erori.

Project DAW ★ Workspace visible Board

**De facut (Razvan)**

- + Add a card

**De facut (Sebi)**

- + Add a card

**De facut (Mihnea)**

- + Add a card

**In lucru (Razvan)**

- + Add a card

**In lucru (Sebi)**

- + Add a card

**In lucru (Mihnea)**

- + Add a card

**Facut (Razvan)**

- Crearea proiectului și instalarea tool-urilor
- Conexiunea la server și baza de date
- + Add a card

**Facut (Sebi)**

- + Add a card

**Facut (Mihnea)**

- fixed Migrations
- + Add a card

**Plan de lucru**

- de implementat un front page în stilul stackoverflow.com
- 3 tipuri de utilizatori: utilizator neînregistrat, înregistrat și administrator
- pe pagina principală avem categorii de discuție
- in fiecare categorie avem mai multe subiecte de discuție
- de acordat drepturile utilizatorilor de a edita și viziona în funcție de rolul lor
- + Add a card

**De facut (Sebi)**

- + Add a card

**De facut (Razvan)**

- + Add a card

**De facut (Mihnea)**

- + Add a card

**Facut (Razvan)**

- Crearea proiectului și instalarea tool-urilor
- Conexiunea la server și baza de date
- + Add a card

**Facut (Sebi)**

- + Add a card

**Facut (Mihnea)**

- fixed Migrations
- + Add a card

**Plan de lucru**

- de implementat un front page în stilul stackoverflow.com
- 3 tipuri de utilizatori: utilizator neînregistrat, înregistrat și administrator
- pe pagina principală avem categorii de discuție
- in fiecare categorie avem mai multe subiecte de discuție
- de acordat drepturile utilizatorilor de a edita și viziona în funcție de rolul lor
- + Add a card

**Plan de lucru**

- de implementat un front page în stilul stackoverflow.com
- 3 tipuri de utilizatori: utilizator neînregistrat, înregistrat și administrator
- pe pagina principală avem categorii de discuție
- in fiecare categorie avem mai multe subiecte de discuție
- de acordat drepturile utilizatorilor de a edita și viziona în funcție de rolul lor
- + Add a card

**De facut (Sebi)**

- + Add a card

**De facut (Razvan)**

- + Add a card

**De facut (Mihnea)**

- + Add a card

**Facut (Razvan)**

- Crearea proiectului și instalarea tool-urilor
- Conexiunea la server și baza de date
- + Add a card

**Facut (Sebi)**

- .gitignore file
- + Add a card

**Facut (Mihnea)**

- fixed Migrations
- + Add a card

**Plan de lucru**

- de implementat un front page în stilul stackoverflow.com
- 3 tipuri de utilizatori: utilizator neînregistrat, înregistrat și administrator
- pe pagina principală avem categorii de discuție
- in fiecare categorie avem mai multe subiecte de discuție
- de acordat drepturile utilizatorilor de a edita și viziona în funcție de rolul lor
- + Add a card

**De facut (Sebi)**

- + Add a card

**De facut (Razvan)**

- + Add a card

**De facut (Mihnea)**

- + Add a card

**Facut (Razvan)**

- Crearea proiectului și instalarea tool-urilor
- Conexiunea la server și baza de date
- C.R.U.D. pe categoriile de discuție
- C.R.U.D. pe discuții
- View-urile pentru home index, categorii și discuții
- + Add a card

**Facut (Sebi)**

- .gitignore file
- + Add a card

**Facut (Mihnea)**

- fixed Migrations
- + Add a card



**Trello Project DAW**

The screenshot shows three identical Trello boards stacked vertically, each representing a different user's perspective or role in the project. The boards are titled "Project DAW" and "Workspace visible". Each board has a "Board" tab selected.

**Plan de lucru** (Leftmost Column):

- De facut (Sebi): Incluzarea unui editor de text
- De facut (Razvan): sortare
- De facut (Mihnea): sistem votare
- Facut (Razvan): Crearea proiectului si instalarea tool-urilor
- Facut (Sebi): .gitignore file
- Facut (Mihnea): fixed Migrations

**De facut (Sebi)** (Second Column):

- + Add a card

**De facut (Razvan)** (Third Column):

- + Add a card

**De facut (Mihnea)** (Fourth Column):

- + Add a card

**Facut (Razvan)** (Fifth Column):

- Conexiunea la serverul pe baza de date
- CR.U.D. pe categoriile de discutii
- View-uri pentru home index, categorii si discutii
- paginare pentru discutiile dintr-o categorie
- Edit / New / Delete pe categoriile de discutii sunt acum conditionate de rolul utilizatorului
- de acordat drepturi utilizatorilor de edit si vizionare in functie de rolul lor
- + Add a card

**Facut (Sebi)** (Sixth Column):

- .gitignore file
- 3 tipuri de utilizatori: utilizator neinregistrat, inregistrat si administrator
- Authorize Roles pentru DiscussionsController
- Authorize Roles pentru CategoriesController
- AnswersController
- C.R.U.D. pe Answer
- CommentsController
- C.R.U.D. pe Comentarii
- de acordat drepturi utilizatorilor de edit si vizionare in functie de rolul lor
- Discussion Show View
- AccountsController -> vizualizare + edit profil utilizator
- View-uri pentru AccountsController
- Incarcare poza de profil
- Sistem notificare
- + Add a card

**Facut (Mihnea)** (Seventh Column):

- fixed Migrations
- better Views with Bootstrap
- Define Votes, stabilirea relatiilor intre tabelele Votes, Discussions si Answer. Implementarea realizarilor
- Aduaugarea Functionalitatii Votes pentru Discussions, stabilirea utilitatii (0 - neutru, 1 upvote, 2 downvote)
- Aduaugarea Functionalitatii Votes pentru Answer, utilizarea este similara cu Discussions
- Highlighting - UI: fiecare buton il va fi punctat/dinsosat in functie de activitatea lui de user pentru Votes
- BugFix
- + Add a card

**De facut (Sebi)** (Second Column):

- + Add a card

**De facut (Razvan)** (Third Column):

- + Add a card

**De facut (Mihnea)** (Fourth Column):

- + Add a card

**Facut (Razvan)** (Fifth Column):

- Conexiunea la serverul pe baza de date
- CR.U.D. pe categoriile de discutii
- View-uri pentru home index, categorii si discutii
- paginare pentru discutiile dintr-o categorie
- 2 sortari de discutii (alfabetic + dupa popularitatea discutiei)
- motor de cautare
- UsersController -> unde admin-ul gestioneaza conturile altor utilizatori
- View-uri pentru UsersController
- + Add a card

**Facut (Sebi)** (Sixth Column):

- .gitignore file
- 3 tipuri de utilizatori: utilizator neinregistrat, inregistrat si administrator
- Authorize Roles pentru DiscussionsController
- Authorize Roles pentru CategoriesController
- AnswersController
- C.R.U.D. pe Answer
- CommentsController
- C.R.U.D. pe Comentarii
- de acordat drepturi utilizatorilor de edit si vizionare in functie de rolul lor
- Discussion Show View
- AccountsController -> vizualizare + edit profil utilizator
- View-uri pentru AccountsController
- Incarcare poza de profil
- Sistem notificare
- Includerea unui editor de text summenotat
- bug stergere in cascada
- maintaining scroll position while adding/editing a comment or
- + Add a card

**Facut (Mihnea)** (Seventh Column):

- fixed Migrations
- better Views with Bootstrap
- Define Votes, stabilirea relatiilor intre tabelele Votes, Discussions si Answer. Implementarea realizarilor
- Aduaugarea Functionalitatii Votes pentru Discussions, stabilirea utilitatii (0 - neutru, 1 upvote, 2 downvote)
- Aduaugarea Functionalitatii Votes pentru Answer, utilizarea este similara cu Discussions
- Highlighting - UI: fiecare buton il va fi punctat/dinsosat in functie de activitatea lui de user pentru Votes
- BugFix
- sortari dupa upvoturi
- cova Views/HTML
- 2 sortari si la raspunsurile dintr-o discutie
- Fixed Bug la sortari
- sistem votare
- + Add a card

**Plan de lucru** (Leftmost Column):

- + Add a card

**De facut (Sebi)** (Second Column):

- + Add a card

**De facut (Razvan)** (Third Column):

- + Add a card

**De facut (Mihnea)** (Fourth Column):

- + Add a card

**Facut (Razvan)** (Fifth Column):

- Crearea proiectului si instalarea tool-urilor
- Conexiunea la serverul pe baza de date
- CR.U.D. pe categoriile de discutii
- View-uri pentru home index, categorii si discutii
- paginare pentru discutiile dintr-o categorie
- 2 sortari de discutii (alfabetic + dupa popularitatea discutiei)
- motor de cautare
- UsersController -> unde admin-ul gestioneaza conturile altor utilizatori
- View-uri pentru UsersController
- + Add a card

**Facut (Sebi)** (Sixth Column):

- .gitignore file
- 3 tipuri de utilizatori: utilizator neinregistrat, inregistrat si administrator
- Authorize Roles pentru DiscussionsController
- Authorize Roles pentru CategoriesController
- AnswersController
- C.R.U.D. pe Answer
- CommentsController
- C.R.U.D. pe Comentarii
- de acordat drepturi utilizatorilor de edit si vizionare in functie de rolul lor
- Discussion Show View
- AccountsController -> vizualizare + edit profil utilizator
- View-uri pentru AccountsController
- Incarcare poza de profil
- Sistem notificare
- Includerea unui editor de text summenotat
- bug stergere in cascada
- maintaining scroll position while adding/editing a comment or
- + Add a card

**Facut (Mihnea)** (Seventh Column):

- fixed Migrations
- better Views with Bootstrap
- Define Votes, stabilirea relatiilor intre tabelele Votes, Discussions si Answer. Implementarea realizarilor
- Aduaugarea Functionalitatii Votes pentru Discussions, stabilirea utilitatii (0 - neutru, 1 upvote, 2 downvote)
- Aduaugarea Functionalitatii Votes pentru Answer, utilizarea este similara cu Discussions
- Highlighting - UI: fiecare buton il va fi punctat/dinsosat in functie de activitatea lui de user pentru Votes
- BugFix
- sortari dupa upvoturi
- cova Views/HTML
- 2 sortari si la raspunsurile dintr-o discutie
- Fixed Bug la sortari
- sistem votare
- + Add a card

The screenshot shows a Trello board titled "Project DAW". The board has several columns representing tasks assigned to different team members:

- Plan de lucru**: Contains cards for "De facut (Sebi)", "De facut (Razvan)", "De facut (Mihnea)", and "Facut (Razvan)".
- De facut (Sebi)**: Contains cards for "De facut (Razvan)", "De facut (Mihnea)", and "Facut (Sebi)".
- De facut (Razvan)**: Contains cards for "De facut (Mihnea)" and "Facut (Razvan)".
- De facut (Mihnea)**: Contains cards for "Facut (Razvan)", "Facut (Sebi)", and "Facut (Mihnea)".
- Facut (Razvan)**: Contains cards for "Create project and install tools", "Connect server to database", "CRUD operations for discussion categories", "CRUD operations for discussions", "View URLs for home index", "More navigation buttons", "Edit / New / Delete for discussion categories", "Pagination for discussions", "2 sorting options (alphabetical + popularity)", "Search function", "UserController -> handle admin roles", "View URLs for UserController", and "+ Add a card".
- Facut (Sebi)**: Contains cards for "Create project and install tools", "Connect server to database", "CRUD operations for discussion categories", "CRUD operations for answers", "CommentsController", "CRUD operations for comments", "Allow users to edit their profile based on role", "Discussion Show View", "AccountsController -> visualize + edit user profile", "View URLs for AccountsController", "Upload user profile picture", "System notifications", "Include a text editor for summernote", "Bug fix", "Highlighting UI", "Fix bug for sorting", "System votare", "Compiler", "Python, C++, Csharp", "CRUD responses contain code", and "+ Add a card".
- Facut (Mihnea)**: Contains cards for "Create project and install tools", "Connect server to database", "CRUD operations for discussion categories", "CRUD operations for answers", "CommentsController", "CRUD operations for comments", "Allow users to edit their profile based on role", "Discussion Show View", "AccountsController -> visualize + edit user profile", "View URLs for AccountsController", "Upload user profile picture", "System notifications", "Include a text editor for summernote", "Bug fix", "Highlighting UI", "Fix bug for sorting", "System votare", "Compiler", "Python, C++, Csharp", "CRUD responses contain code", and "+ Add a card".

