# 

# **SQUAD MANAGER**

# **★ SPORT TEAM MANAGEMENT APP ★**

# 

# 

# **Developed by**

# **↳ Dijmărescu Cristina**

# **↳ Haiducu Ștefan**

# **↳ Ionescu Alexandru**

# **↳ Stan Ana-Maria**

# **Descriere aplicație:**

“Squad Manager” este o aplicatie web pentru gestionarea unui club sportiv. Aceasta este adresata in mod special cluburilor care isi desfasoara activitatea in campionatele de fotbal.

Aplicatia a fost dezvoltata special pentru a usura munca cluburilor sportive, ajutandu-i pe oamenii ce activeaza in industria sportiva sa gestioneze mai facil clubul, sa aiba acces oricand la toate informatiile echipei.

Squad Manager, dupa cum sugereaza si denumirea, preia din atributiile managerului clubului. Memorand toate datele importante de la nivelul clubului, aplicatia este oriand pregatita sa iti ofere orice informatie ai nevoie referitoare la echipa ta. Cu aplicatia Squad Manager ai un asistent pregatit in permanenta sa raspunda cautarilor tale.

Interfata aplicatiei este formata din 7 pagini web ce permit utilizatorilor sa vizualizeze date despre despre clubul sportiv.

* Pagina HOME:

Pagina de acasa

* Pagina PLAYERS:

In aceasta pagina utilizatorii pot vizualiza date despre toti jucatorii care activeaza la club. Pentru fiecare jucator aplicatia prezinta numele, prenumele, nationalitatea, data nasterii, inaltimea, piciorul de executie, pozitia de joc si valoarea de piata. Prin accesarea acestei pagini antrenorii si membrii staffului pot oricand sa primeasca o imagine de ansamblu asupra lotului care este disponibil in cadrul echipei.

* Pagina STAFF:

In aceasta pagina utilizatorii pot vizualiza date despre toti membrii staffului care lucreaza la club. Pentru fiecare membru din staff aplicatia prezinta numele, prenumele, rolul in cadrul echipei, data nasterii, emailul si telefonul de contact. Prin accesarea acestei pagini utilizatorii pot oricand sa primeasca o imagine de ansamblu asupra intregului staff al echipei care se afla la dispozitia clubului.

* Pagina PLAYERS CONTRACT:

In aceasta pagina utilizatorii pot vizualiza date despre contractele care au fost semnate de jucatorii transferati la club. Pagina ofera o privire de ansamblu completa asupra situatiilor contabile si financiare ale clubului in ceea ce priveste angajamentele pe care firma le are cu jucatorii sai, oferind detalii despre data semnarii si data expirarii contractelor, salariul negociat si agentul FIFA.

* Pagina STAFF CONTRACT:

In aceasta pagina utilizatorii pot vizualiza date despre contractele care au fost semnate de catre membrii staffului care au fost angajati de catre club. Pagina ofera o privire de ansamblu completa asupra situatiilor contabile si financiare ale clubului in ceea ce priveste angajamentele pe care acesta le are cu stafful, oferind detalii despre data semnarii si data expirarii contractelor, salariul negociat si agentul FIFA.

* Pagina MATCHES:

In aceasta pagina utilizatorii pot vizualiza date despre meciurile pe care echipa le-a disputat sau le are programate pentru a se disputa in viitor. Pagina ofera date complete despre fiecare partida de joc, specificandu-se stadionul pe care evenimentul a avut loc, oponentul impotriva caruia s-a desfasurat meciul, competitia, data evenimentului, scorul partidei si numele arbitrului care a oficializat disputa dintre cele doua echipe.

* Pagina STADIUMS:

In aceasta pagina utilizatorii pot vizualiza date despre stadioanele pe care clubul le are la dispozitie pentru a desfasura eventimente sportive. Pagina ofera date complete despre fiecare stadion din tara sau din strainatate pe care echipa a disputat partide, fiecare stadion pe care echipa il detine in baza de antrenament sau fiecare stadion pe care echipa planuieste sa desfasoare meciuri. Pentru fiecare stadion din aceasta pagina se pot vizualiza numele acestuia, capacitatea locurilor, suprafata de joc si adresa sa.

# **User stories**

# Ca orice utilizator, imi doresc sa pot crea un cont cu user si parola in care sa ma pot loga de fiecare data cand accesez platforma, pentru o mai buna securitate

* Ca orice utilizator, imi doresc, atunci cand ma inregistrez in platforma sa imi fie atribuit un rol in cadrul clubului (admistrator, antrenor, utilizator simplu), pentru o mai buna securitate.

# Ca orice utilizator, vreau sa pot verifica contractele de munca care au fost inregistrate in contabilitate pentru a ma asigura ca totul este in deplina legalitate

# Ca orice utlizator, vreau sa pot vizualiza date despre fiecare meci pe care echipa l-a disputat, pentru a putea discuta strategii de joc.

# Ca orice utilizator, imi doresc sa pot accesa date despre toate stadioanele pe care a jucat echipa noastra sau pe care le avem noi in baza sportiva

* Ca orice utilizator, vreau sa pot vizualiza informatii de baza despre jucatorii si staff ul auxiliar ce activeaza in cadrul echipei
* Ca orice utilizator, vreau sa pot sorta crescator si descrescator dupa orice criteriu listele cu jucatori, staff, stadioane, contracte, meciuri pentru a vizualiza mai rapid datele care ma intereseaza pe moment.
* Ca orice utilizator, imi doresc sa am acces la diferite statistici importane pentru club, cum ar fi salariul mediu, câți jucători are clubul pe fiecare post, câți membri ai personalului, înălțimea medie pe post etc, pentru a avea o privire de ansamblu asupra activitatii echipei

# Ca utilizator administrator, imi doresc sa am acces deplin la orice functie din platforma, sa pot modifica,adauga, sterge orice element doresc din baza de date, pentru a manageria cat mai bine echipa

* Ca utilizator administrator, imi doresc sa am in fiecare pagina un formular unde sa pot introduce manual un jucator nou, un membru al staffului, un stadion, un meci sau un contract.

# Ca utilizator administrator, imi doresc sa am un formular in paginile pentru juctori, staff si contracte unde sa pot prelua date de pe site-ul oficial de fotbal Transfermarket, pentru a le aduce in aplicatia mea.

* Ca utilizator administrator, imi doresc sa am in fiecare pagina, in dreptul fiecarui element din lista, optiunea de a sterge sau a modifica acel element.
* Ca utilizator antrenor, imi doresc sa am disponibile butoane de update si delete in paginile pentru jucatori, staff, meciuri si stadioane pentru a putea opera modificari in cadrul echipei.
* Ca utilizator antrenor, imi doresc sa nu pot modifica in cadrul aplicatiei date ce tin de staff-ul tehnic sau de contractele staff-ului, pentru a evita posibile erori.
* Ca utilizator simplu, imi doresc sa am acces doar in a vizualiza paginile aplicatiei, fara a putea opera modificari pe acestea, evitand astfel modificari aduse din greseala sau in mod intentionat de catre persoane neautorizate.

Mai multe date despre cerintele din user story pot fi accesate pe linkurile:

<https://yana5.atlassian.net/jira/software/projects/PMB/boards/2>

<https://yana5.atlassian.net/jira/software/projects/PROIECT/boards/3>

# 

# 

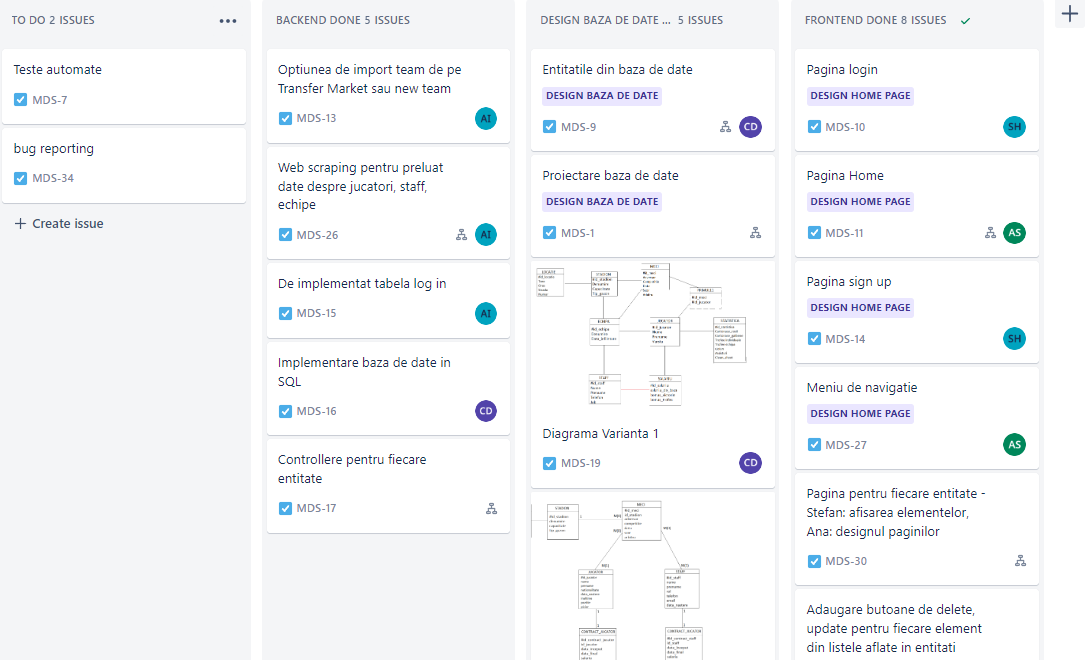
# 

# 

# 

# **Backlog creation**

Ne-am monitorizat progresul pe backend, frontend si design-ul bazei de date folosind board-ul Jira. Mai departe, intrarile din tabele au comentarii si branch-uri care trateaza mai amanuntit implementarea sau rezolvarea problemei care se discuta. Acesta poate fi gasit la linkul: <https://yana5.atlassian.net/jira/software/projects/MDS/boards/1>



### 

### 

### 

### **Features list**

* Pagina manageriere jucatori care activeaza la echipa
* Pagina manageriere membri staffului din cadrul echipei
* Pagini managerierei contracte de munca pentru angajatii din echipa
* Pagina manageriere meciuri disputate in sezonul competitional
* Pagina manageriere stadioane disponibile in cadrul clubului
* Formular in pagina pentru introducere link Transfermarket de unde aplicatia sa preia automat datele despre jucatori/staff
* Formular cu campuri specifice pentru introducere element nou, disponibil in toate sectiunile aplicatiei
* Butoane pentru stergere sau modificare date disponibile pentru fiecare element din lista fiecarei sectiuni din aplicatie
* Sortari crescatoare/descrescatoare/alfabetice disponibile pentru listele din toate sectiunile aplicatiei
* Formular pentru inregistrare cont nou utilizator
* Formular de log in pentru utilizatorii deja existenti
* Optiune de log out inainte de parasirea aplicatiei
* Statistici despre activitatiile clubului

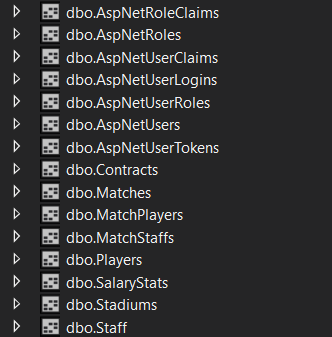
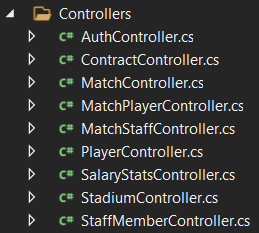
### **Arhitectura, Design**

Intrucat aplicatia Squad Manager se bazeaza pe stocarea unor cantitati mari de date, am decis sa utilizam in implementarea acesteia o baza de date de tip SQL realizata in backend si transpusa apoi vizual in frontend.

**‣Backend:**

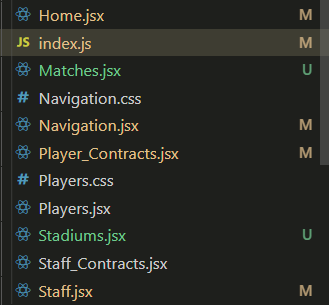
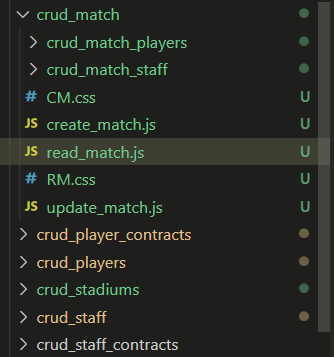
Partea de backend s-a realizat cu ajutorul .net si C# implementate in cadrul aplicatiei Visual Studio 2019. Baza de date din spatele aplicatiei cuprinde 5 tabele principale (player, staffmember, contract, match, stadium), 2 tabele bridge care leaga entitatile ce au o relatie many to many (matchplayer, matchstaff) si un tabel separat (salarystats) pe care dorim sa il utilizam in afisarea diverselor statistici in cadrul clubului. Operatiile CRUD la nivelul bazei de date s-au realizat cu ajutorul controllerelor deschise pentru fiecare entitate in parte, in care s-au creat metode specifice de get, post, delete, update.

Pe langa controllerele specifice entitatilor ce compun elementele unui club de fotbal, a fost necesar si implementarea serviciului de autentificare si inregistrare useri, definind astfel si un controller specific autentificarii.

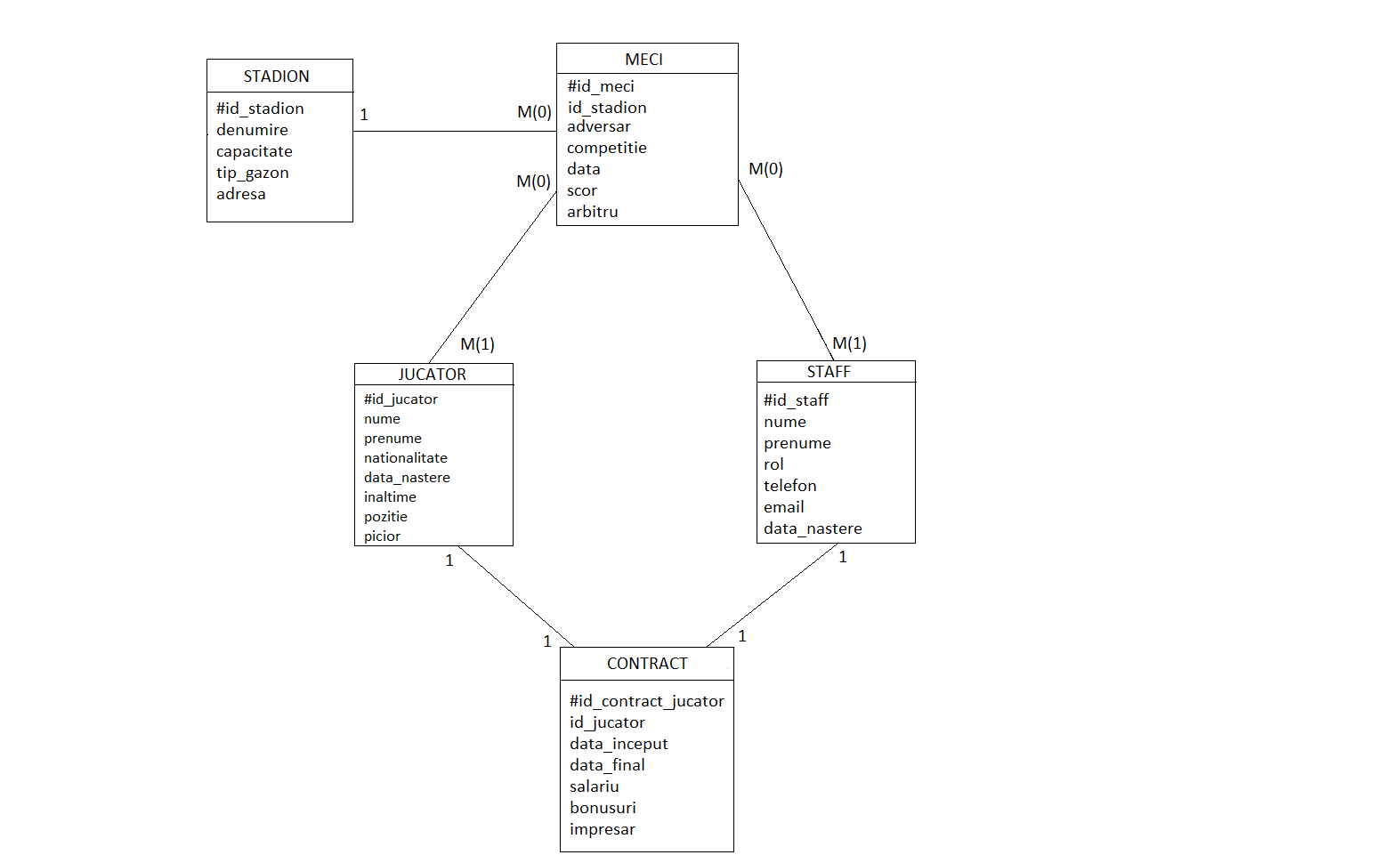
**‣Frontend:**

Partea de frontend s-a realizat cu ajutorul HTML, CSS, Java Script, React implementate in cadrul aplicatiei Visual Studio Code. Pentru fiecare entitate din baza de date s-a creeat vizual o pagina, astfel utilizatorii putand sa acceseze oricand informatiile ce se afla stocate in baza intr-o maniera user-friendly, usor de utilizat. Fiecare pagina se acceseaza dintr-un meniu simplu de navigare aflat in partea de sus a aplicatiei. Pentru a face accesibile de catre utilizatori metodele de CRUD implementate in backend, am creeat componente CRUD pentru fiecare entitate, care utilizeaza metodele definite in .net .

Arhitectura bazei de date consta in 5 entitati principale prezente in diagrama ER.

In diagrama conceptuala, intre tabelele jucator si meci, va aparea o entitate de tip “bridge table” care va coordona relatial many to many dintre cele doua entitati avand drept atribute id-urile celor doua. Intre tabelele staff si meci, vom avea de asemenea implementata o entitate aditionala care va gestiona relatia many to many, avand atribute id-urile celor doua tabele.



Pentru a introduce date in baza se pot utiliza formularele prezente in fiecare pagina html dezvoltata in frontend (introducere manuala) sau pentru tabelele jucator si contractele aferente acestora, datele se pot prelua cu ajutorul unor linkuri de pe site-ul Transfermarket. Linkul se introduce in campul special prezent in paginile care permit preluarea datelor din linkuri, iar apoi datele sunt incarcate in baza de date.

Pentru a prelua corect datele trebuie ca linkul sa fie preluat doar de pe Transfermarket, din pagina specifica unei echipe, din sectiunea Detailed.

EXEMPLU LINK: <https://www.transfermarkt.com/fc-barcelona/kader/verein/131/saison_id/2021/plus/1>

**Source control and bugs**

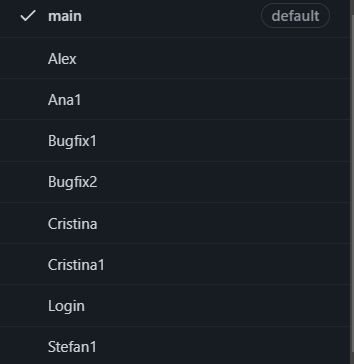
Pentru a putea lucra eficient, am folosit Github pentru a ne sincroniza munca si a o putea separa pe branch-uri, dar si pentru a putea raporta bug-uri. Fiind impartiti pe doua grupe, una lucrand la backend si una la frontend, am creat doua repository-uri, unul pentru fiecare grupa.

### **‣ Repo-ul backend**

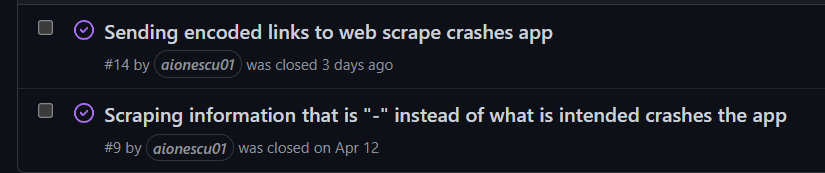
****

Dupa cum se poate vedea, acest repo are 46 de commit-uri inainte de finalizarea proiectului si 15 merge-uri in branch-ul main.

Cateva din branch-urile folosite sunt urmatoarele:



De asemenea, au fost cateva bug-uri pe care nu le-am putut rezolva imediat, astfel ca am lansat un issue, iar cu ajutorul celorlalti, acestea au fost rezolvate.



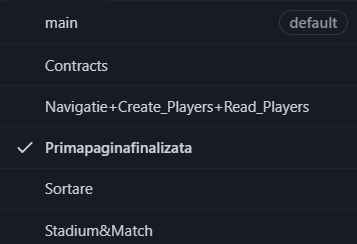
### **‣Repo-ul frontend**

### **Commit-uri 1 frontend**

**Commit-uri 2 frontend**

Dupa cum se poate vedea, acest repo are 13 commit-uri inainte de finalizarea proiectului.

Cateva din branch-urile folosite sunt urmatoarele:

****

Acest repo are de asemenea mai multe branch-uri cu un numar mare de commit-uri, dar si bug-uri care au fost eventual rezolvate.



# 

# 

# **Integration testing**

In cazul programului nostru, unit testing-ul nu este foarte relevant, intrucat componentele functioneaza impreuna. De exemplu, toate functiile au nevoie de tabelele din baza de date pentru a functiona, iar baza de date nu poate fi testata singura. Astfel, folosim integration testing, acesta evaluand componentele pe un nivel mai larg, verificand daca componentele functioneaza impreuna.

Fiind destul de asemanatoare, daca o functie functioneaza, suntem aproape siguri ca si restul functiilor de acelasi tip functioneaza. Testam, deci, cate o functie de fiecare tip: una pentru POST, una pentru GET, una pentru PUT, una pentru DELETE si o functie POST cu link pentru web scraping.



# **Utilizare aplicatie:**

Pentru a testa aplicatia va trebui sa colonati backendul si frontendul

Link github frontend: <https://github.com/aionescu01/Frontend-MDS>

Link github backend: <https://github.com/aionescu01/Proiect-MDS>

Se deschide Visual Studio 2019 si se incarca fisierul cu proiectul de backend, iar apoi se apasa butonul IIS Express pentru activare.

Se deschide Visual Studio Code si se incarca fisierul cu proiectul de frontend. Se deschide un terminal nou, se utilizeaza comanda cd ./proiect/ si apoi comanda npm start, dupa care aplicatia va rula.