

Specifikacija projekta - Elektronski Dnevnik

"Elektronski dnevnik" je sistem namenjen efikasnom praćenju i upravljanju školskim aktivnostima. Ovaj sistem omogućava nastavnicima, učenicima i roditeljima pristup relevantnim informacijama o ocenama, prisustvu, rasporedu časova i drugim školskim aktivnostima. Cilj sistema je poboljšanje komunikacije između svih učesnika u obrazovnom procesu

Projekat ima sledeće zadatke:

- upravljanje podacima
- izveštavanje
- bezbednost

Upravljanje podacima

Entiteti

U sistemu (web aplikaciji) postoje četiri tipa korisnika: administratori, nastavnici, učenici i roditelji

Korisnik je opisan sledećim podacima:

- id
- lozinka
- korisničko ime(možda - concat(ime, ".", prezime)) !!?
- uloga u sistemu (moguće vrednosti su: ROLE_ADMIN, ROLE_TEACHER, ROLE_STUDENT i ROLE_PARENT)

Posebni/Dodatni podaci:

ROLE_TEACHER:

- ime
- prezime
- email
- Predmeti koje predaje
- razredni (povezivanje sa Odeljenje)
- konsultacije

ROLE_STUDENT:

- ime
- prezime
- datum rođenja
- Odeljenje
- Roditelj (povezivanje sa ROLE_PARENT)
- Lista Predmeta

ROLE_PARENT:

- ime
- prezime
- email
- br. tel
- adresa
- Lista dece (povezivanje sa ROLE_STUDENT)

Pored korisnika, aplikacija rukuje sa sledećim entitetima:

- Predmet
- Ocena
- Odeljenje
- Razred i
- dodatni(raspored, škola, polugodište, prisutnost...)

Predmet

- id
- naziv
- nedeljni fond časova

- Razred
- lista: ROLE_TEACHER

Odeljenje

- id
- naziv
- Razred
- školska godina
- Lista Predmeta

Razred

- 1-8

Ocena(slabi)

- 1-5
- datum dodele
- ocenjivački podaci: ROLE_TEACHER, ROLE_STUDENT, Predmet, vrsta(pismeni, usmeni...), komentar !!?

Veze između entiteta:

Nastavnik predaje neke predmete nekim učenicima. Nastavnik može ali ne mora biti razredni nekom odeljenju. Nastavnik može predavati više predmeta, dok jedan predmet može imati više nastavnika. Učenik može iz nekog predmeta u nekom polugodištu nekog razreda imati više ocena (od 1 do 5) koje daje nastavnik koji taj predmet predaje tom učeniku. Učenik pripada određenom razredu i odeljenju. Učenik može pripadati više razreda tokom školovanja, s obzirom na eventualno promenjena odeljenja. Predmet se održava u nekom broju razreda. Svaki student i svaki nastavnik odgovaraju po jednom korisniku. Roditelj može imati više dece, dok jedno dete može imati više roditelja. Odeljenje ima više učenika. Odeljenje ima više predmeta. Ocene su povezane sa određenim nastavnikom, učenikom, predmetom i datumom

Lista pod-zadataka i slučajeva korišćenja:

- Zadatak treba da omogući, na početku, da se Korisnik uloguje u sistem koristeći korisničko ime i lozinku
- Da Korisnik može da promeni lozinku
- Korisnik može biti: Nastavnik, Učenik, Administrator, Roditelj
- Ako je korisnik Učenik, može da vidi sve svoje Ocene
- Ako je korisnik Roditelj, može da vidi sve Ocene svih Učenika vezanih za sebe u svim Školama
- Ako je korisnik Roditelj, može da vidi sve Predmete svih Učenika vezanih za sebe u svim Školama
- Ako je korisnik Nastavnik, može da vidi sve Ocene svih svojih Predmeta za Učenike i Predmete kojima predaje i da ih menja, briše, ili dodaje nove uz mogućnost pretrage
- Ako je korisnik Nastavnik(razredni) može imati potpun pristup svim podacima u sistemu za svoj razred
- Ako je korisnik Administrator može imati potpun pristup svim podacima u sistemu

Dužnosti administratora:

- Dodavanje, brisanje, i izmena: Predmeta, Nastavnika, Učenika, Roditelja, Ocena
- Dodavanje i izmena uključuje sva obeležja i sve veze entiteta koji se dodaje i menja.
-

Izveštaji:

- Log datoteke čuvaju sve značajne događaje tokom rada servera
- Samo administrator može pregledati i preuzeti logove
- Roditelju se šalje e-mail kada učenik dobije Ocenu(posle 24h), sa informacijama o Oceni

Bezbednost:

Aplikacija treba da podrži rad samo sa prijavljenim korisnicima (Učenici, Nastavnici, Roditelji i Administratori) Svaki od korisnika ima tačno određene akcije koje sme da obavlja

Potrebno je kreirati sledeće REST endpoint-e:

/korisnik:

GET: Dobavi sve korisnike - lista(/project/users) + dobavi po id(project/users/{id})

□□□- u slučaju da ne postoji korisnik sa prosleđenim id-em vratiti grešku

POST: Kreiraj novog korisnika (project/users)

□□□- vraća kreiranog korisnika

PUT: Ažuriraj podatke korisnika(putanja project/users/{id})

- Ukoliko je prosleđen ID koji ne pripada nijednom korisniku metoda treba da vrati □□grešku, a u suprotnom vraća podatke korisnika sa izmenjenim vrednostima
- NAPOMENA: u okviru ove metode ne menjati vrednost atributa uloga i lozinka
- (atribut ulogu ima pravo samo administrator da promeni)
- DELETE: Obriši korisnika (project/users/{id})
- metoda je rezervisana za administratora, tačnije samo on ima pravo da obriše korisnike
- ukoliko je prosleđen ID koji ne pripada nijednom korisniku metoda treba da vrati □□grešku, a u suprotnom vraća obrisano korisnika
- REST endpoint za logovanje - prijavu na sistem (project/users/login)
- metoda treba da vrati ulogovanog korisnika, odnosno grešku ako ne postoji □□korisnik sa datim kredencijalima
- REST endpoint za odlogovanje - odjavu sa sistema (project/users/logout)
- metoda treba da vrati odlogovanog korisnika, odnosno grešku ako ne postoji □□ulogovan korisnik
- REST endpoint koji omogućava izmenu vrednosti atributa uloga postojećeg korisnika □□(/project/users/changeUser/{id}/role/{role})
- metoda je rezervisana samo za administratora
- ukoliko je prosleđen ID koji ne pripada nijednom korisniku metoda treba da vrati □□grešku, a u suprotnom vraća podatke korisnika sa izmenjenom vrednošću atributa uloga
- REST endpoint koji omogućava izmenu vrednosti atributa lozinka postojećeg korisnika □□(/project/users/changePassword/{id})
- ukoliko je prosleđen id koji ne pripada nijednom korisniku metoda treba da vrati □□grešku, a u suprotnom vraća podatke korisnika sa izmenjenom vrednošću atributa lozinka
- metodi se prosleđuju stara i nova lozinka, a da bi vrednost atributa lozinka mogla □□da bude zamenjena sa novom vrednošću, neophodno je da se vrednost stare □□lozinke korisnika poklapa sa vrednošću prosleđene stare lozinke
- REST endpoint koji vraća korisnika po vrednosti prosleđenog username-a □□(/project/users/by-username/{username})
- u slučaju da ne postoji korisnik sa prosleđenim username-om vratiti grešku
- REST endpoint koji dodaje Predmet Korisniku(Nastavnik, Učenik) □□(/project/users/{userId}/subject/{subjectId})
- u slučaju da ne postoji korisnik ili predmet sa prosleđenim id vratiti grešku
- metoda je rezervisana samo za administratora ili razrednog

/predmet:

- GET: Dobavi sve predmete - lista(project/subjects) + dobavi po id(project/subjects/{id})
- u slučaju da ne postoji predmet sa prosleđenim id-em vratiti grešku
- POST: Kreiraj novi predmet (project/subjects)
- vraća kreirani predmet
- metoda je rezervisana za administratora i razrednog
- PUT: Ažuriraj podatke predmeta(project/subjects/{id})
- Ukoliko je prosleđen ID koji ne pripada nijednom predmetu metoda treba da vrati □□grešku, a u suprotnom vraća podatke predmeta sa izmenjenim vrednostima
- DELETE: Obriši predmet (project/subjects/{id})
- metoda je rezervisana za administratora, tačnije samo on ima pravo da obriše predmete
- ukoliko je prosleđen ID koji ne pripada nijednom predmetu metoda treba da vrati □□grešku, a u suprotnom vraća obrisani predmet
- REST endpoint koji vraća predmet po vrednosti prosleđenog naziva □□(/project/subjects/by-subjectName/{subjectName})
- u slučaju da ne postoji predmet sa prosleđenim naziv-om vratiti grešku
- REST endpoint koji vraća predmet po vrednosti prosleđenog nastavnika □□(/project/subjects/by-teacher/{teacherId})
- u slučaju da ne postoji predmet sa prosleđenim nastavnikom vratiti grešku
- REST endpoint koji vraća predmet po vrednosti prosleđenog učenika + ocene □□(/project/subjects/by-student/{studentId})
- u slučaju da ne postoji predmet sa prosleđenim učenikom vratiti grešku
-

/odeljenje:

- GET: Dobavi sva odeljenja - lista(project/classes), dobavi po id - Imenik(project/classes/{id})
- u slučaju da ne postoji odeljenje sa prosleđenim id-em vratiti grešku

POST: Unesi novo odeljenje(project/classes)

□□- vraća kreirano odeljenje

PUT: Ažuriraj podatke odeljenja(project/classes/{id})

□□- Ukoliko je prosleđen ID koji ne pripada nijednom odeljenju metoda treba da vrati

□□□grešku, a u suprotnom vraća podatke odeljenja sa izmenjenim vrednostima

DELETE: Obriši odeljenje (project/classes/{id})

□□- metoda je rezervisana za administratora, tačnije samo on ima pravo da obriše predmete

□□- ukoliko je prosleđen ID koji ne pripada nijednom odeljenju metoda treba da vrati

□□□grešku, a u suprotnom vraća obrisano odeljenje

□□REST endpoint koji vraća odeljenja po vrednosti prosleđenog razreda

□□□(/project/classes/by-gradeLevel/{gradeLevel})

□□- u slučaju da ne postoji predmet sa prosleđenim razred-om vratiti grešku

□□REST endpoint za dodavanje predmeta odeljenju(/project/classes/{classId}/subject/{subjectId})

□□- u slučaju da ne postoji odeljenje ili predmet sa prosleđenim id vratiti grešku

□□- metoda je rezervisana samo za administratora ili razrednog

□□REST endpoint za dodavanje učenika odeljenju(/project/classes/{classId}/users/{userId})

□□- u slučaju da ne postoji odeljenje ili korisnik sa prosleđenim id vratiti grešku

□□- metoda je rezervisana samo za administratora ili razrednog

□□REST endpoint koji dobavlja predmete po odeljenju - lista(/project/classes/{classId}/subject/{subjectId})

□□- u slučaju da ne postoji odeljenje ili predmet sa prosleđenim id vratiti grešku

/ocena:

GET: Dobavi sve ocene

POST: Unesi novu ocenu(project/grades/{subjectId}/{studentId})

□□- vraća kreiranu ocenu

PUT: Ažuriraj podatke ocene(project/grades/{subjectId}/{studentId})

□□- Ukoliko je prosleđen ID koji ne pripada nijednom predmetu ni učeniku metoda treba da vrati

□□□grešku, a u suprotnom vraća podatke ocene sa izmenjenim vrednostima

DELETE: Obriši ocenu(project/grades/{subjectId}/{studentId})

□□REST endpoint za pronalazak svih ocena određenog učenika(/project/grades/findByStudent/{studentId})

□□- Ukoliko je prosleđen ID koji ne pripada nijednom učeniku metoda treba da vrati

□□□grešku, a u suprotnom vraća podatke

□□

□/prisutnost:

GET: Dobavi sve podatke o prisutnosti

POST: Unesi novu prisutnost

PUT: Ažuriraj podatke o prisutnosti

DELETE: Obriši podatke o prisutnosti□

Bonus funkcionalnosti:

- Dodati entitet Polugodište kako bi se omogućilo praćenje ocena po polugodištima
- Implementirati mogućnost praćenja prisustva učenika na časovima
- izveštaji za: Nastavnike, Školu, o uspehu učenika(polugodište - kraj)
- Nastavnik, Roditelj, Učenik (Raspored časova + raspored Pisanih provera)
- Pisana provera(Predmet, Datum, lista Učenika Odeljenja, Ocene...)
- Učenik (vladanje, aktivnost)
- Nastavnik (upis časa + odsustvo Učenika)
- Nastavnik-razredni, Roditelj, Učenik (izostanci)
- Omogućiti Administratoru/Nastavniku da vrši rezervaciju termina za roditeljske sastanke