

DAB: Projectopdracht 2023-2024

Het project voor dit vak bestaat uit **drie delen** met elk een aparte deadline. Op die manier is er ruimte voor tussentijdse feedback. Hieronder volgt een overzicht van de delen, die in de volgende secties worden beschreven:

Deel	Opdracht	Deadline
1.	EER-schema (formatief)	Maandag 25/03/2024 (week voor Paasvakantie)
2.	EER-schema + SQLite database	Maandag 22/04/2024 (2e maandag na Paasvakantie)
3.	Java-applicatie	Een week voor het examen (TBD)

Het is een **groepsproject** waarbij je samenwerkt per 2. Als het aantal studenten oneven is, mag één groep per 3 samenwerken.

Indienen deel 1 Upload je EER-schema en een kort verslag (zie hieronder) op Toledo.

(Vermeld in je verslag de namen van al je teamleden)

Indienen deel 2 Upload je verbeterde EER-schema en je `.sqlite`` bestand (+eventueel verslag) (zie hieronder) op Toledo.

Indienen deel 3 Exporteer je hele JavaFX-project (zie hieronder) naar een zip en upload op Toledo.

Voor elke indiening geldt dat je per groep slechts één upload moet doen.

De opdracht

Tennis Vlaanderen organiseert **toernooien** over heel Vlaanderen. De voorzitter van Tennis Vlaanderen is op zoek naar bekwame programmeurs die software kunnen realiseren om de administratie van de organisatie te ondersteunen.

Met de software moet je uiteraard **tennisspelers** kunnen aanmaken. Elke tennisspeler heeft een globale ID bij Tennis Vlaanderen die gebruikt wordt als identifier wanneer een persoon zich inschrijft voor een toernooi. Een speler moet op zijn Tennis Vlaanderen-dashboard een globaal overzicht kunnen opvragen van het aantal gespeelde matches, het aantal gewonnen matches, verloren matches en de hoogst bereikte plaats in een toernooi (bv. Finalist T.C.Ham reeks enkel Heren t.e.m. 5 punten OF Winnaar G.T.Tessenderlo reeks enkel Dames t.e.m. 10 punten). Naast de fysieke parameters van een tennisspeler zoals lengte, gewicht, leeftijd, geslacht, ...), moet de applicatie ook de ranking van elke tennisspeler bijhouden. Deze ranking is een waarde van 1 tot en met 10 waarbij 1 de laagste ranking is en 10 de hoogste.

Daarnaast kan een tennisspeler ook een rol als wedstrijdleader en/of scheidsrechter opnemen.

- Een **wedstrijdleader** regelt de planning van het toernooi, is het aanspreekpunt voor vragen en doet op de wedstrijddagen de inschrijvingen van de spelers.
Aan elk toernooi moet minstens één **wedstrijdleader** gekoppeld zijn.
- Aan elke finalewedstrijd van een toernooi (= halve finales en grote + kleine finale) moet één **scheidsrechter** toegewezen worden.

Je mag als uitbreiding ook nog andere rollen toevoegen die aan een toernooi of een wedstrijd gekoppeld worden.

Elk toernooi vindt plaats op een bepaalde **Tennisclub** en bestaat uit minstens **4 reeksen** waarvoor je kan inschrijven (Enkel Heren t.e.m. 5 punten, enkel Heren t.e.m. 10 punten, enkel Dames t.e.m. 5 punten, enkel Dames t.e.m. 10 punten). Een speler kan altijd voor een hogere reeks inschrijven maar

niet voor een lagere (bv. speler met 3 punten kan inschrijven in de reeks t.e.m. 10 punten, maar een speler met 6 punten kan niet inschrijven voor de reeks t.e.m. 5 punten).

Een toernooi wordt georganiseerd volgens het principe van rechtstreekse uitschakeling: de winnaar gaat door naar de volgende ronde; voor de verliezer eindigt het toernooi. Dit proces loopt door tot er per wedstrijdreeks één winnaar overblijft.

Indien het aantal spelers in een wedstrijdreeks geen match van twee is, worden in de eerste ronde zoveel spelers vrijgesteld als nodig. Zij kwalificeren zich rechtstreeks voor de tweede ronde. Dit wordt per loting bepaald.

Alvast bedankt voor de samenwerking en groetjes van de voorzitter van *Tennis Vlaanderen*!

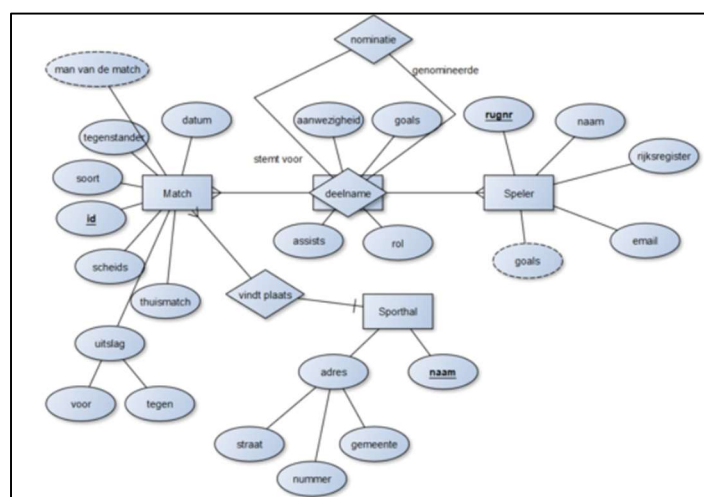


** De tekst hierboven bevat de minimale vereisten van het programma. Hiermee kan je maximaal 16/20 halen voor het project. Je bent dus vrij om dit voorstel uit te breiden. Correct uitgewerkte uitbreidingen kunnen voor een hogere score zorgen, maar enkel als aan de minimale vereisten voldaan is. Hou echter rekening met de werklast en met het feit dat je ook nog op de andere opleidingsonderdelen moet slagen.*

Deel 1

Deadline: zie boven

Voor dit deel verwachten we dat je een **EER-schema** tekent en indient. Dit schema bevat (1) entiteiten/modellen en (2) relaties tussen deze modellen, zoals jij de opdracht interpreteert. Bijvoorbeeld:



Minimum vereisten:

- Alle entiteiten en relaties die nodig volgens de basisopgave.
- De essentiële attributen per entiteit en relatie.

Het schema zou bespreekbaar moeten zijn met eender welk lid van *Tennis Vlaanderen*, ook met mensen zonder een technische achtergrond.

Naast het schema dien je ook een **verslag** in---van ongeveer 500 woorden---waarin je je schema beschrijft én verklaart waarom je denkt dat bepaalde relaties nodig zijn. Het formaat van het verslag is een Word document. Op die manier kunnen we gemakkelijk feedback geven via de 'review' functionaliteit.

Indienen doe je via Toledo; voeg alle nodige bestanden (schema én verslag). Zie boven.

Deze deadline dient enkel ter formatieve feedback. Na feedback wordt er verwacht dat je in deel 2 aan de hand van de feedback het EER-schema nog aanpast en de aangepaste versie terug indient bij deel 2. Enkel die versie van je EER-schema zal dan op punten staan.

Deel 2

Deadline: zie boven

Voor dit deel verwachten we twee onderdelen

- Het verbeterde EER-schema. Integreer de feedback op je eerste ontwerp. Het verslag moet je enkel aanpassen en opnieuw uploaden wanneer fundamenteel veranderd werd
- De uitgewerkte database. Dit mag zowel in SQLite-formaat zijn als ontwikkeld in phpmyadmin met dan een export naar .sql.

Minimum vereisten:

- Uitwerking van het hele schema uit deel 1 in tabellen
- Minstens 2 constraints gebruiken
- Alle tabellen zijn voorzien van de nodige Primary en Foreign Keys
- Voeg een beperkte set van (fake) gegevens toe in de databases (bijvoorbeeld een drietal per tabel)

Denk ook goed na over eventuele (foute) ingaven van gegevens---welke soort data verwacht je waar?

Indienen doe je via Toledo; voeg alle nodige bestanden toe: het verbeterde EER-schema en het '.sqlite' of .sql-bestand. Zie boven.

Deel 3.

Deadline: zie boven

Voor dit deel verwachten we dat je het ontwerp van de database integreert in een **softwareprogramma**, uitgewerkt in Java. Op dit moment maak je best de overstap naar een sqlite-database (als dat nog niet gebeurd is). Er zal een startproject voorzien worden, de UI wordt ontwikkeld in JavaFX (dit staat vast).

Minimum vereisten:

- Gebruik JavaFX als UI.
- Gebruik als toegang naar de SQLite-embedded database JDBC, JDBI, of JPA/Hibernate.

Dit project wordt mondeling verdedigd in de week van het examen. We verwachten dat je de keuze van JDBC vs. JDBI vs. JPA goed kan motiveren!

Starten kan vanaf het JavaFX-enabled startproject: Te downloaden als zip van Toledo. **Volgt nog**

Merk op dat het over het vak "databases" gaat, en we dus NIET de nadruk leggen op de UI, maar eerder op de backend. Denk echter wel opnieuw goed na over de invoer van gegevens.

Indienen doe je via Toledo; export je JavaFX project naar een zip zoals we tijdens de les gezien hebben. Zie boven.

ALVAST VEEL SUCCES!