

Verantwoordingsdocument Eindopdracht App VRC

Tijdens het ontwikkelen van de voetbalclub-app heb ik veel geleerd over programmeren, het structuren van code en het werken met externe API's. In dit document reflecteer ik op mijn proces, benoem ik uitdagingen en geef ik aan welke best practices ik heb toegepast.

1. Reflectie op Twee Problemen en Oplossingen:

Probleem 1: Toegangsrestricties per rapport.

Uitdaging:

Het correct afschermen van rapporten zodat alleen de invuller, TC en HJO deze kunnen zien, bleek complex. In eerste instantie was de toegangscontrole alleen op basis van rol, niet per individueel rapport. Hierdoor konden sommige gebruikers rapporten inzien die niet voor hen bedoeld waren.

Aanpak en Oplossing:

Ik heb de rolpermissies verder aangescherpt in de 'roles.py'-module en onderzocht hoe ik bij het opvragen van rapporten een extra check kon toevoegen op basis van zowel rol als gebruiker. Door te werken met een combinatie van gebruikersnaam en rol bij het tonen van rapporten, is de privacy beter gewaarborgd. Dit proces heeft me geleerd om niet alleen op rol te vertrouwen, maar ook op specifieke gebruikersrechten.

Probleem 2: Synchronisatie van teams en spelers via de Sportslink API:

Uitdaging:

Het synchroniseren van teams en spelers tussen de Sportslink API en de lokale database leverde problemen op, vooral bij het bijwerken en verwijderen van bestaande records. In het begin werden wijzigingen niet goed gelogd en was het lastig te zien welke data was toegevoegd, aangepast of verwijderd.

Aanpak en Oplossing:

Ik heb de synchronisatiefuncties ('sync_teams_with_db' en 'sync_players_with_db' in 'database.py') uitgebreid zodat ze per wijziging een logregel aanmaken. Hierdoor werd het proces inzichtelijker en kon ik eenvoudig controleren of de lokale database overeenkwam met de gegevens uit de API. Ook heb ik foutafhandeling toegevoegd voor netwerkproblemen, zodat de gebruiker duidelijke feedback krijgt bij fouten.

2. Doorontwikkelingen en Verbeterpunten:

Verbeterpunt 1: Modulairder maken van de code.

Toelichting:

Hoewel de app al modulair is opgezet, zijn sommige functies nog te of behandelen ze meerdere verantwoordelijkheden. Het verder opsplitsen van functies en het toepassen van Single Responsibility Principle zou de onderhoudbaarheid en uitbreidbaarheid vergroten.

Waarom wenselijk?

Kleinere goed afgebakende functies zijn makkelijker te testen, te hergebruiken en te gebruiken. Dit voorkomt bugs en maakt het eenvoudiger om nieuwe functionaliteit toe te voegen.

Verbeterpunt 2: Overstappen op een centrale database.

Toelichting:

Momenteel worden alle gegevens opgeslagen in JSON-bestanden. Dit werkt prima voor een enkelvoudige gebruiker, maar is niet schaalbaar als meerdere mensen tegelijk de app gebruiken.

Waarom wenselijk?

Een centrale database (zoals SQLite of PostgreSQL) zorgt voor betere data-integriteit, maakt gelijktijdig gebruik mogelijk en biedt meer mogelijkheden voor rapportages en analyses.

3. Best Practices in de Code.

DRY (Don't Repeat Yourself):

. Voorbeeld:

De synchronisatiefuncties voor teams, spelers en wedstrijden zijn generiek opgezet. Ze gebruiken dezelfde structuur en logica, waardoor codeherhaling wordt voorkomen.

. Resultaat:

Dit maakt het eenvoudiger om aanpassingen door te voeren; een verbetering in één functie kan direct worden doorgevoerd in andere.

Foutafhandeling en Logging:

.Voorbeeld:

Bij het ophalen van data uit de Sportslink API wordt gecontroleerd op netwerkfouten en foutcodes. Wijzigingen worden gelogd in CSV-bestanden, zodat altijd terug te zien is wat er is aangepast.

. Resultaat:

De gebruiker krijgt duidelijke feedback bij fouten en het loggen vergroot de transparantie van het proces.

Modulaire opbouw:

. Voorbeeld:

Functionaliteiten zijn opgesplitst in aparte modules zoals 'database.py', 'matches.py', 'scouting.py' en 'roles.py'. Elke module heeft een eigen verantwoordelijkheid.

. Resultaat:

Dit maakt de code overzichtelijk, onderhoudbaar en eenvoudig uit te breiden.

Toegangscontrole:

. Voorbeeld:

Door het gebruik van rollen en permissies in 'roles.py' wordt de toegang tot modules en rapporten geregeld.

. Resultaat:

Dit verhoogt de veiligheid en zorgt ervoor dat de gebruikers alleen toegang hebben tot relevante onderdelen.

Tot slot:

Het ontwikkelen van deze app heeft me veel geleerd over het oplossen van programmeerproblemen en het toepassen van best practices. Door kritisch te reflecteren op mijn keuzes en uitdagingen ben ik in staat geweest om de kwaliteit van mijn code te verbeteren en een solide basis te leggen voor verdere ontwikkeling.