

Stap 5: Proof of Concept

Werking programma

Bij de ontwikkeling van mijn programma heb ik gewerkt met een gebruiksvriendelijke interface. Tijdens het ontwikkelen ben ik tegen verschillende problemen aangelopen. Een belangrijke les die ik hieruit heb getrokken, is dat ik in de toekomst een "God Class" wil implementeren. Dit zou de code overzichtelijker maken en het aanpassen van functies eenvoudiger maken.

Momenteel heb ik ervoor gekozen om een aparte klasse te maken die verantwoordelijk is voor het beheren van informatie. Dit maakt het eenvoudiger om vanuit één centrale plek wijzigingen aan te brengen, zonder dat ik in verschillende bestanden moet zoeken en aanpassen. Deze opzet draagt bij aan een betere onderhoudbaarheid en zorgt ervoor dat de applicatie gemakkelijker uit te breiden is.

Verder heb ik ervoor gezorgd dat de kernfunctionaliteiten van het programma correct werken en dat gebruikers intuïtief door de applicatie kunnen navigeren.

Testresultaten

In het conceptueel model hebben we verschillende userstories gemaakt, nu is het tijd om dit om te zetten naar een testplan. Doet het programma daadwerkelijk wat we verwacht?

Use case	Invoer	Verwachte uitvoer	Voldaan	Niet voldaan
Betaling verwerken	Als ik op pagina van de leden klik wil ik alle leden zien met hun betaling-status	Ik verwacht dat als ik alle leden in de tabel te zien zijn en hun status zichtbaar is en bijgewerkt wordt als er voldaan wordt aan de betaling	JA	
Speelschema bekijken	Als ik bij wedstrijden kijk wil ik het speelschema zien dat up-to-date is	Ik wil het actuele wedstrijdschema zien en de gegevens die daarbij horen	JA	
Wedstrijd toevoegen	Als ik bij wedstrijd inplannen kijk wil ik een wedstrijd kunnen inplannen die rekening houdt met beschikbaarheid	Zodra ik een wedstrijd wil toevoegen houdt hij rekening met de wedstrijden die al ingepland zijn en werk hij de database bij	JA	

