Software Design Document

Door: Renas Khalil en Sadek al Mousawi
Groep: Groep 2 RS
Datum: 07 Oktober 2024
Versie: 1.2
Vak: Cross-Platform Development (24/25)

Inhoudsopgave

1. Requirements

1.1 Functionele eisen

Dit zijn de vereisten die beschrijven wat het systeem moet kunnen doen. Elk van deze eisen is direct gerelateerd aan de functies die het systeem moet bieden om aan de gebruikerswensen te voldoen.

Registreren

REQ-001: Het systeem moet gebruikers in staat stellen zich te registreren met een geldig e-mailadres en wachtwoord.

Inloggen

REQ-002: Het systeem moet geregistreerde gebruikers in staat stellen in te loggen met een e-mailadres en wachtwoord.

REQ-003: Het systeem moet ondersteuning bieden voor wachtwoordherstel via e-mail.

Teams aanmaken voor evenementen

REQ-004: Gebruikers moeten teams kunnen aanmaken en beheren, inclusief het aanpassen van teamnaam en beschrijving voor evenementen.

REQ-005: Het systeem moet teamleiders in staat stellen om teamleden specifieke rollen toe te wijzen (bijv. beheerder of deelnemer).

Deelnemen aan teams voor evenementen

REQ-006: Gebruikers moeten via een link of QR-code kunnen deelnemen aan een team.

REQ-007: Het systeem moet automatische meldingen sturen naar teamleiders wanneer nieuwe leden zich aansluiten.

Gebruikers uitnodigen via link of QR-code

REQ-008: Het systeem moet gebruikers de mogelijkheid bieden om anderen uit te nodigen via een unieke link of QR-code.

Route plannen

REQ-009: Het systeem moet gebruikers in staat stellen routes te plannen voor teamactiviteiten, inclusief starttijd en locatie.

REQ-010: Het systeem moet integreren met kaartdiensten zoals Google Maps voor het plannen van routes.

Roosters tonen

REQ-011: Het systeem moet een overzichtelijk rooster tonen met geplande teamactiviteiten.

REQ-012: Gebruikers moeten meldingen kunnen ontvangen van aankomende activiteiten.

Persoonlijk rooster

REQ-013: Het systeem moet een persoonlijk rooster bieden, zodat gebruikers hun activiteiten kunnen bekijken, ongeacht het team waarbij ze zijn aangesloten.

REQ-014: Als een gebruiker lid is van meerdere teams binnen één evenement, moet het evenement maar één keer worden weergegeven op hun persoonlijk rooster.

User Roles

REQ-015: Het systeem moet gebruikers verschillende rollen kunnen toewijzen, zoals beheerder, deelnemer, en gast.

REQ-016: Beheerders moeten exclusieve toegang hebben tot bepaalde functies zoals teambeheer en het uitnodigen van leden.

1.2 Niet-functionele eisen

Dit zijn de vereisten die beschrijven wat het systeem beter, veiliger, of betrouwbaarder maakt. Ze hebben betrekking op de kwaliteit en prestaties van het systeem en kunnen worden beoordeeld op basis van de ISO 25010 norm.

Prestaties (Performance Efficiency)

REQ-017: Het systeem moet kunnen omgaan met meerdere gelijktijdige gebruikers zonder prestatieverlies.

REQ-018: De responstijd van het systeem moet geoptimaliseerd zijn.

Beveiliging (Security)

REQ-019: Alle gebruikersgegevens moeten versleuteld worden opgeslagen.

Gebruiksvriendelijkheid (Usability)

REQ-020: Het systeem moet toegankelijk zijn voor gebruikers met een visuele beperking.

REQ-021: Het systeem moet een intuïtieve gebruikersinterface hebben met minimale training vereist.

Onderhoudbaarheid (Maintainability)

REQ-022: De broncode van het systeem moet modulair zijn, zodat nieuwe functies gemakkelijk kunnen worden toegevoegd zonder de bestaande functionaliteit te beïnvloeden.

REQ-023: Het systeem moet een uitgebreide testdekking hebben, met minimaal code coverage door unit tests.

Overdraagbaarheid (Portability)

REQ-024: Het systeem moet compatibel zijn voor browser, applicatie en mobiele applicatie.

2. Functioneel Ontwerp

2.1 Doel van het Systeem

Het systeem is ontworpen om teamsamenwerking te faciliteren door gebruikers in staat te stellen teams te maken, lid te worden, en samen te werken via routes en roosters. De belangrijkste functionaliteiten van het systeem zijn gericht op gebruiksgemak, toegankelijkheid en samenwerking tussen gebruikers.

2.2 Functionele Structuur van het Systeem

Registreren: Gebruikers kunnen een account aanmaken door middel van een e-mailadres en wachtwoord. Het systeem biedt validatie van e-mailadres en het instellen van beveiligingsmaatregelen zoals wachtwoordherstel.

Inloggen: Gebruikers kunnen inloggen met hun geregistreerde gegevens.

Teams aanmaken: Gebruikers hebben de mogelijkheid om teams aan te maken en deze te beheren. Teamleiders kunnen teamnamen instellen, beschrijvingen toevoegen en een teamlogo uploaden.

Deelnemen aan teams: Gebruikers kunnen via uitnodigingen lid worden van teams. Dit kan door het klikken op een unieke link of het scannen van een QR-code die door de teamleider wordt gedeeld.

Gebruikers uitnodigen via link of QR code: Teamleiders kunnen teamleden uitnodigen via een unieke link of een QR-code die door het systeem gegenereerd wordt.

Route plannen: Gebruikers kunnen routes plannen voor teamactiviteiten, zoals routes voor evenementen of vergaderingen. Hierbij worden tijd en locatie gespecificeerd, en het systeem biedt integratie met kaarten (zoals Google Maps) voor navigatie.

Roosters tonen: Het systeem toont de geplande activiteiten in een overzichtelijk roosterformaat. Dit stelt gebruikers in staat om makkelijk inzicht te krijgen in aankomende taken, vergaderingen en routes.

User Roles: Het systeem ondersteunt meerdere gebruikersrollen, zoals beheerder, lid en gast. Elke rol heeft verschillende toegangsrechten en mogelijkheden binnen teams.

2.3 Volledigheid van Requirements

Om er zeker van te zijn dat alle requirements aanwezig zijn, wordt het volgende gecontroleerd:

Geen ontbrekende requirements: Elke functionaliteit die in de requirements staat, is gedetailleerd uitgewerkt in dit FO. Alle voorgestelde features moeten volledig worden gedekt door het systeem.

Realisatie van gebruikerswensen: Er wordt beoordeeld of alle gebruikerswensen met betrekking tot eenvoud, flexibiliteit, en efficiëntie van het systeem worden behaald. Denk bijvoorbeeld aan het gemak van het uitnodigen van nieuwe teamleden of het gebruik van de routeplanner.

2.4 Controle op Gebruikerswensen

Om te bepalen of het systeem alle gebruikerswensen vervult:

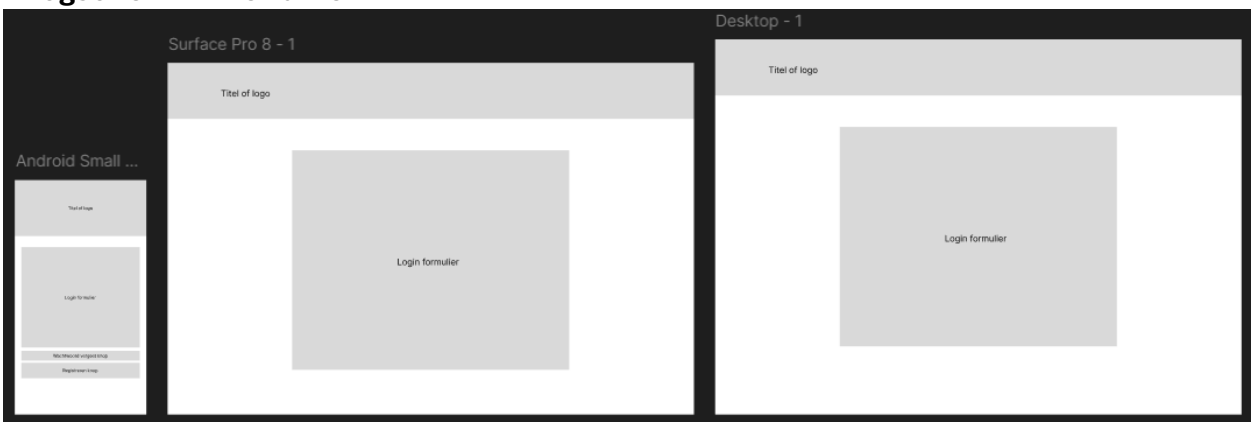
Gebruiksgemak: Het systeem moet eenvoudig en intuïtief te gebruiken zijn, zowel voor nieuwe gebruikers als voor ervaren teamleiders.

Flexibiliteit: Het systeem moet genoeg flexibiliteit bieden voor verschillende soorten teams en activiteiten. Denk aan meerdere rollen, de mogelijkheid om teams aan te passen, en het delen van informatie via links of QR-codes.

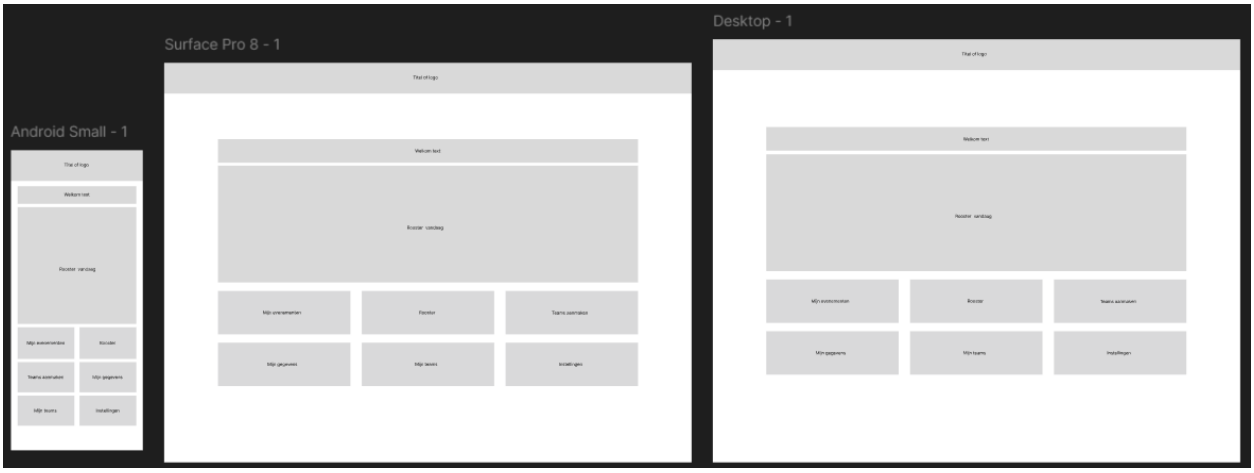
Schaalbaarheid: Het systeem moet schaalbaar zijn, zodat het kan groeien met een toenemend aantal gebruikers en teams zonder het performance te verlagen.

2.5 Wireframes

Inlogscherm Wireframe



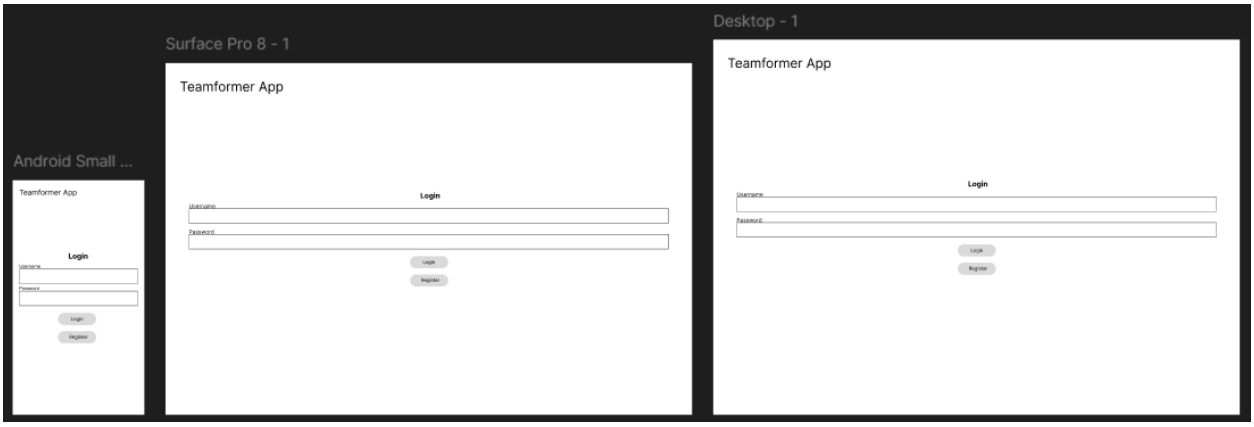
Homepagina Wireframe



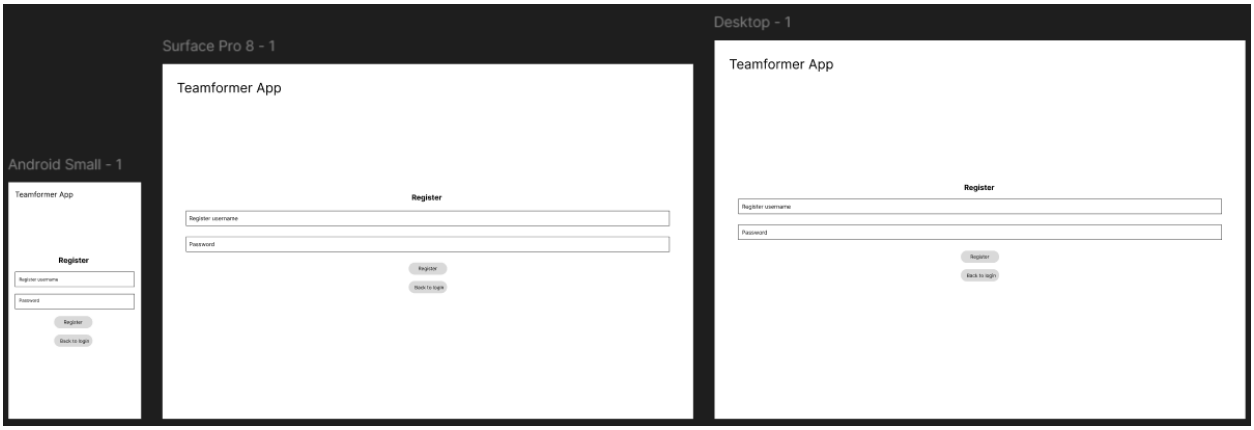
Voordat we begonnen met het ontwikkelen van de applicatie, hebben we eerst het design uitvoerig besproken met de opdrachtgever. We hebben de projectdefinitie geanalyseerd en logisch nagedacht over hoe we dit op een gebruiksvriendelijke manier konden vertalen naar een optimale gebruikerservaring. Deze ideeën zijn vervolgens uitgewerkt in wireframes en, na validatie door de opdrachtgever, verder ontwikkeld tot gedetailleerde mock-ups.

2.6 Mockups

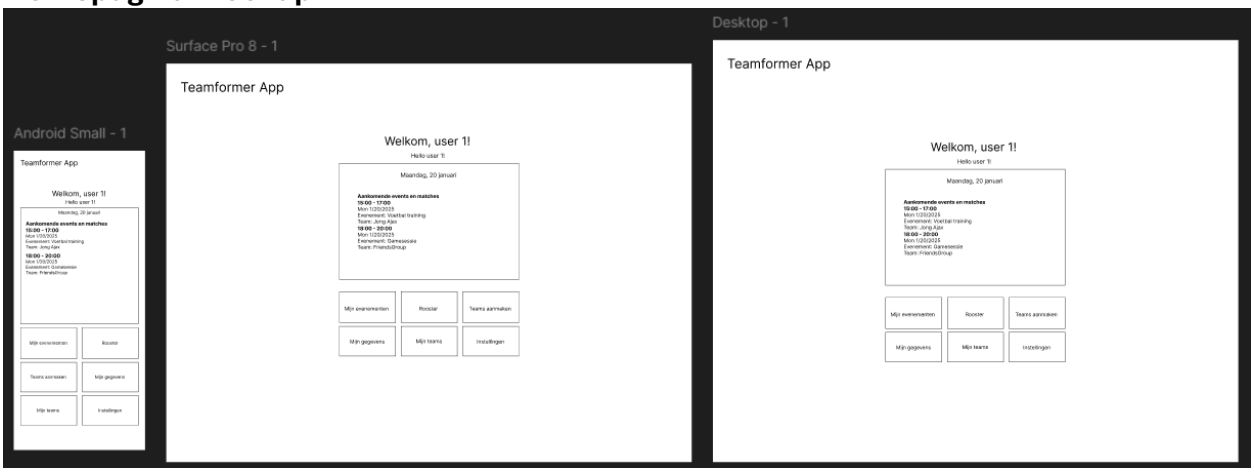
Login pagina mockup



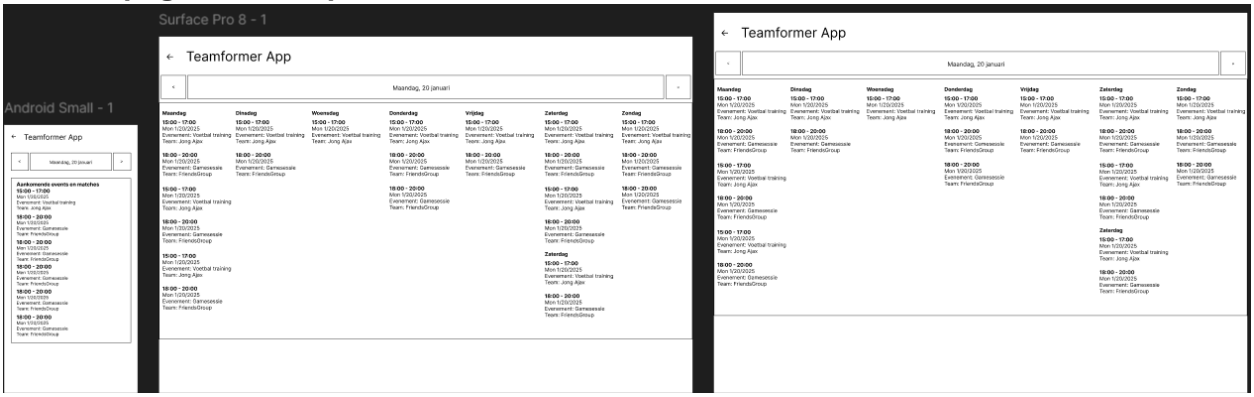
Registratie pagina mockup



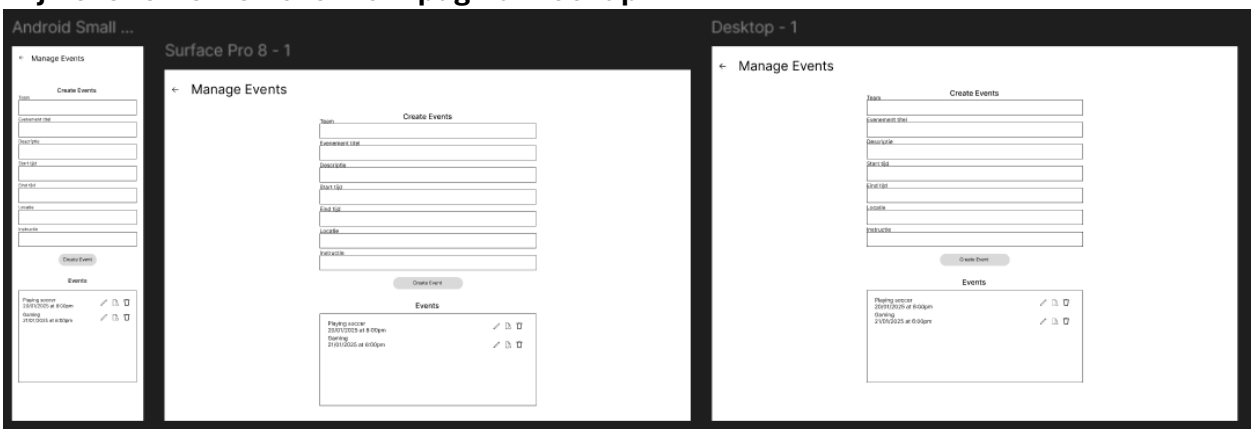
Homepagina mockup



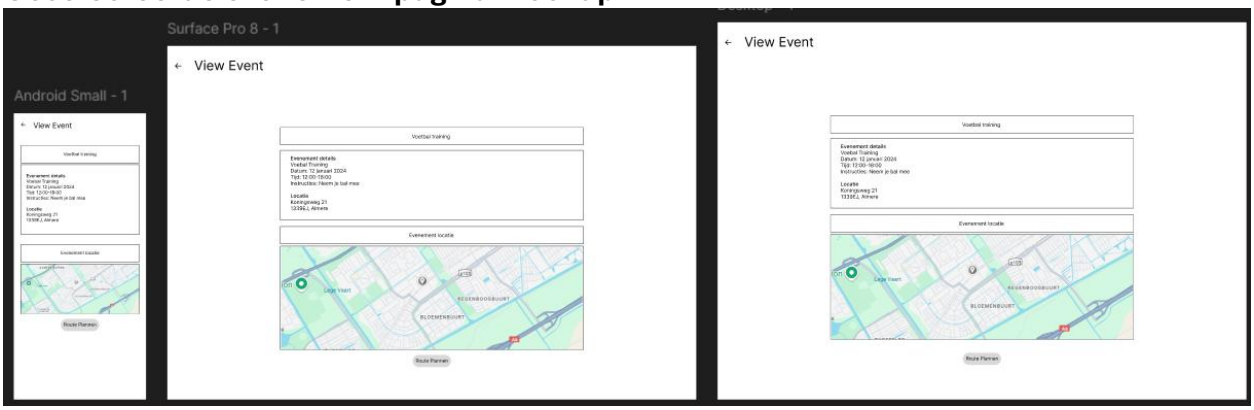
Rooster pagina mockup



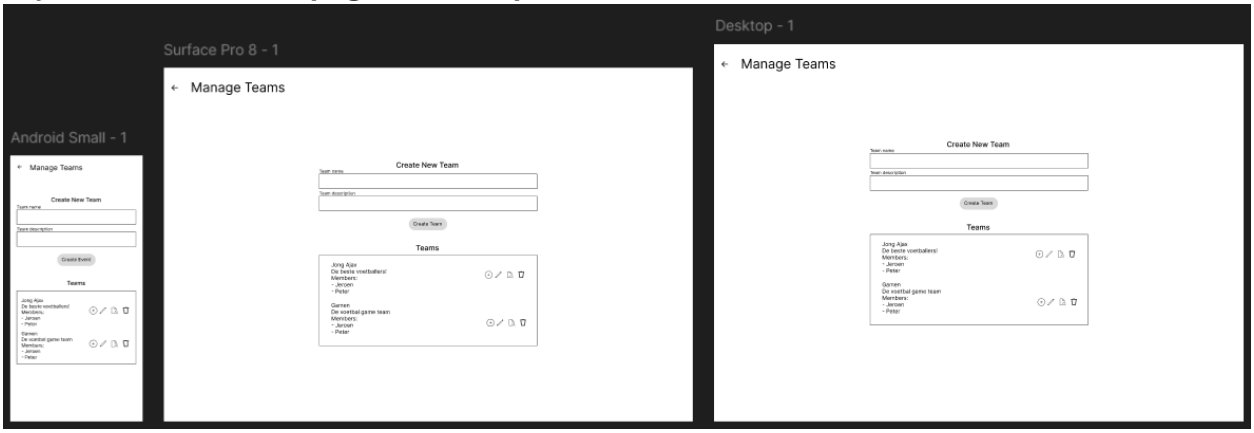
Mijn evenementen overzicht pagina mockup



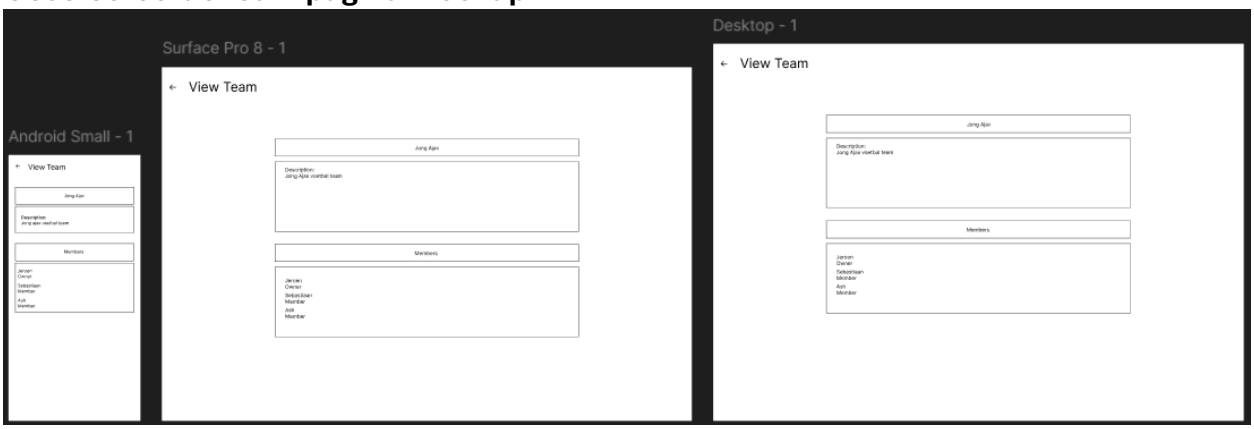
Geselecteerde evenement pagina mockup



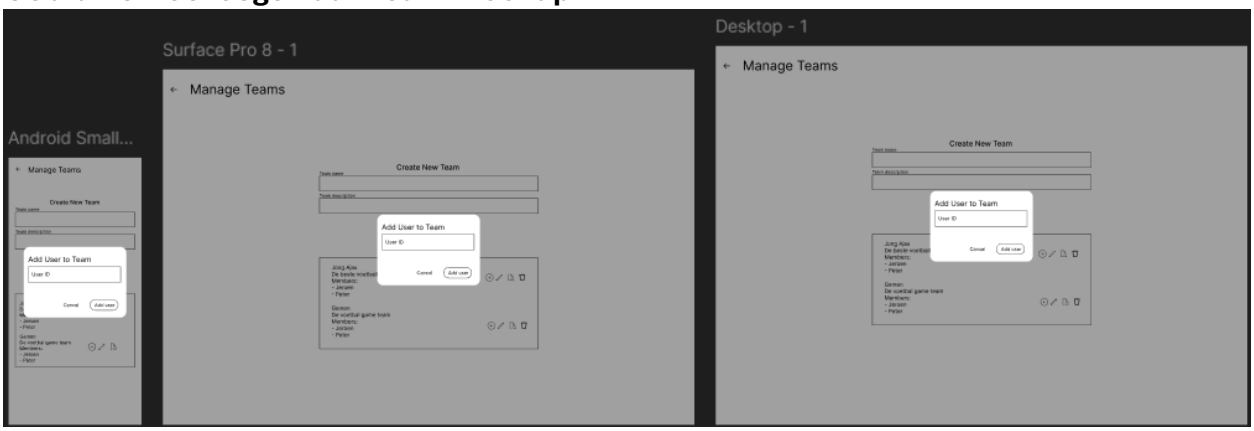
Mijn teams overzicht pagina mockup



Geselecteerde team pagina mockup



Gebruiker toevoegen aan team mockup



3. UML

UML

UML staat voor Unified Modeling Language. Het is een gestandaardiseerde modelleertaal die wordt gebruikt om het gedrag en de architectuur van softwaresystemen te visualiseren, specificeren, ontwerpen en documenteren. UML maakt gebruik van verschillende soorten diagrammen om dit doel te bereiken.

Source: https://www.lucidchart.com/pages/nl/wat-is-unified-modeling-language

Wij maken gebruik van UML-diagrammen voor het vastleggen van de structuur en het gedrag van het systeem.

In ons cross-platformproject maken we gebruik van de volgende diagrammen:

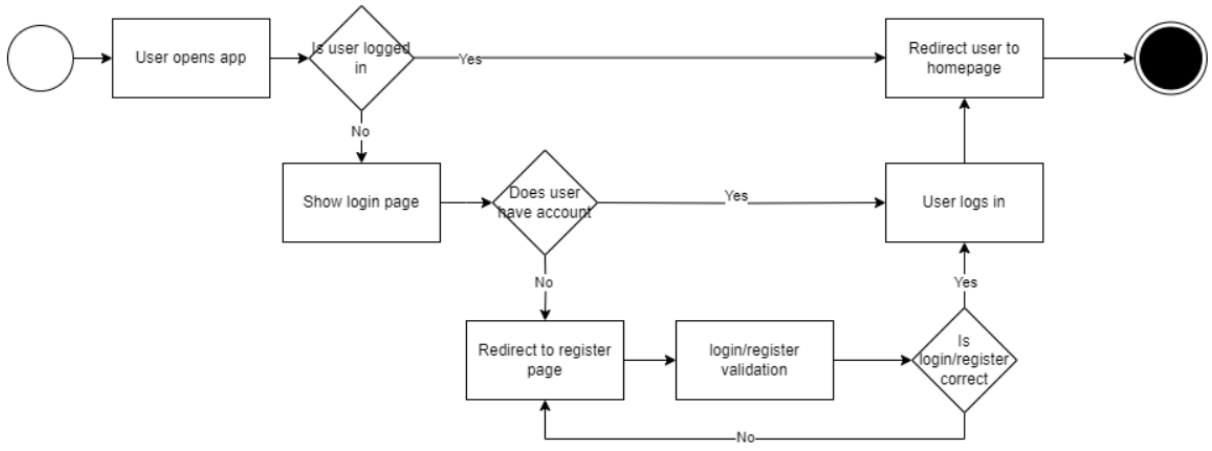
1. Activity diagram
2. Use Case Diagram
3. Sequence Diagram

3.1 Activity Diagram

Activity diagram: Activiteitenschema is een diagram dat het verband weergeeft tussen verschillende activiteiten of bewerkingen meestal van een proces binnen een bedrijf.

Gebruiken opend de applicatie en logt in:

De gebruiker logt in en navigeerd naar de homepagina.



User opens app

Dit is de initiële stap in het diagram waarbij de gebruiker de applicatie opent. Hier zijn geen specifieke functionele eisen aan gekoppeld, maar het vormt de basis van de gebruikersstroom.

1. **Is user logged in? (REQ-002)**

Deze stap checkt of de gebruiker al ingelogd is.

REQ-002: Het systeem moet geregistreerde gebruikers in staat stellen in te loggen met een e-mailadres en wachtwoord.

Hier wordt gecontroleerd of de gebruiker al een sessie heeft. Zo niet, wordt deze doorverwezen naar de loginpagina.

2. **Show login page (REQ-002, REQ-003)**

Als de gebruiker nog niet is ingelogd, wordt de loginpagina weergegeven.

REQ-002: Het systeem moet geregistreerde gebruikers in staat stellen in te loggen met een e-mailadres en wachtwoord.

REQ-003: Het systeem moet ondersteuning bieden voor wachtwoordherstel via e-mail.

Deze stap faciliteert het inloggen en biedt, indien nodig, een herstelmogelijkheid.

3. **Does user have an account? (REQ-001)**

Hier moet de gebruiker aangeven of hij al een account heeft. Als dit niet het geval is, wordt de gebruiker doorverwezen naar de registratiepagina.

REQ-001: Het systeem moet gebruikers in staat stellen zich te registreren met een geldig e-mailadres en wachtwoord.

Dit vereiste zorgt ervoor dat nieuwe gebruikers een account kunnen aanmaken om de applicatie te gebruiken.

4. **Redirect to register page (REQ-001)**

Als de gebruiker nog geen account heeft, wordt hij doorgestuurd naar de registratiepagina.

REQ-001: Het systeem moet gebruikers in staat stellen zich te registreren met een geldig e-mailadres en wachtwoord.

De registratiepagina maakt het mogelijk dat de gebruiker een account kan aanmaken met de benodigde gegevens.

5. **Login/Register validation (REQ-002, REQ-003)**

Het systeem valideert de ingevoerde inlog- of registratiegegevens.

REQ-002: Het systeem moet geregistreerde gebruikers in staat stellen in te loggen met een e-mailadres en wachtwoord.

REQ-003: Het systeem moet ondersteuning bieden voor wachtwoordherstel via e-mail.

Deze validatie voorkomt dat onjuiste gegevens worden geaccepteerd en biedt ondersteuning bij verkeerde invoer.

6. **User logs in (REQ-002)**

Als de inloggegevens correct zijn, wordt de gebruiker ingelogd.

REQ-002: Het systeem moet geregistreerde gebruikers in staat stellen in te loggen met een e-mailadres en wachtwoord.

Dit bevestigt dat de gebruiker succesvol is geverifieerd en toegang krijgt tot de applicatie.

7. **Is login/register correct? (REQ-004, REQ-005, REQ-006)**

Deze stap bepaalt of de ingevoerde gegevens correct zijn.

REQ-004, REQ-005, REQ-006: Hoewel deze requirements niet direct te maken hebben met het inlogproces, vormen ze een basis voor het correct toewijzen van rollen en rechten aan de gebruiker, wat essentieel is bij login en registratie.

Als de gegevens correct zijn, wordt de gebruiker doorgestuurd. Zo niet, dan moet hij terug naar de registratiepagina of loginpagina.

8. **Redirect user to homepage (REQ-013)**

Zodra de gebruiker succesvol is ingelogd, wordt hij doorgestuurd naar de homepage.

REQ-013: Het systeem moet een persoonlijk rooster bieden, zodat gebruikers hun activiteiten kunnen bekijken, ongeacht het team waarbij ze zijn aangesloten.

Hier moet de applicatie navigatie bieden naar de juiste onderdelen van de applicatie op basis van de gebruikersstatus.

Koppeling met de niet-functionele requirements

Prestaties (REQ-017, REQ-018)

Het systeem moet kunnen omgaan met meerdere gelijktijdige gebruikers zonder prestatieverlies en de responstijd moet geoptimaliseerd zijn. Dit is relevant in het gehele login- en registratieproces, vooral tijdens validatiestappen (Login/Register validation) waar meerdere gelijktijdige aanroepen kunnen plaatsvinden.

1. **Beveiliging (REQ-019)**

Alle gebruikersgegevens moeten versleuteld worden opgeslagen. Dit is van toepassing bij de stappen waarbij gebruikersgegevens worden ingevoerd, gevalideerd en opgeslagen (bijv. Login/Register validation).

2. **Gebruiksvriendelijkheid (REQ-020, REQ-021)**

De login- en registratiepagina's moeten toegankelijk zijn voor gebruikers met een visuele beperking en een intuïtieve interface hebben (Show login page, Redirect to register page).

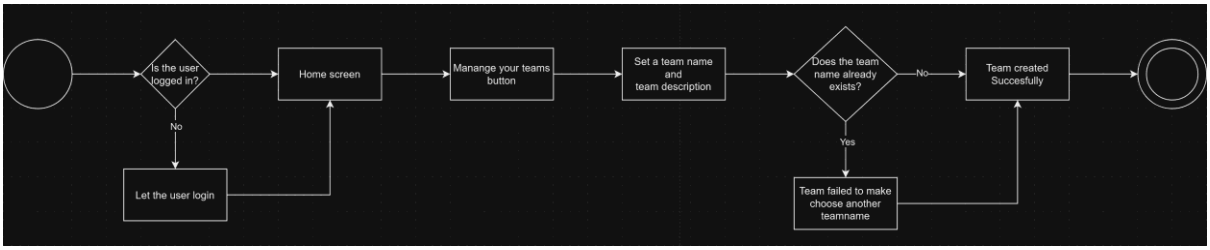
3. **Onderhoudbaarheid (REQ-022, REQ-023)**

De broncode moet modulair zijn en er moet een uitgebreide testdekking zijn. De modulaire opzet van het login- en registratieproces kan hieraan bijdragen, zodat wijzigingen in het registratieproces niet het inlogproces beïnvloeden.

4. **Overdraagbaarheid (REQ-024)**

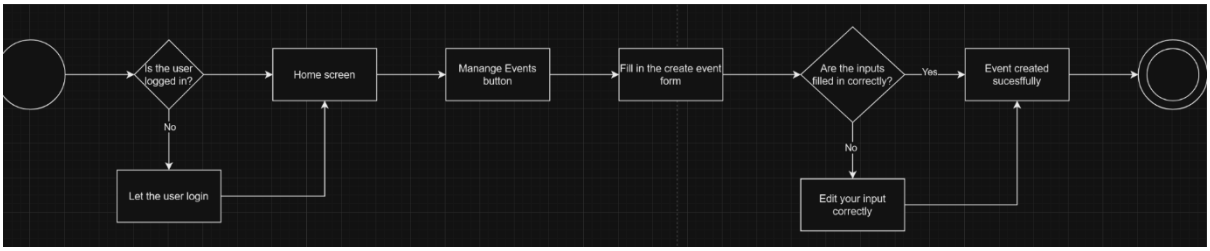
Compatibiliteit met verschillende platformen is belangrijk voor het inlogproces, zodat gebruikers ongeacht het apparaat toegang hebben (User opens app, Show login page).

Team Management Diagram: Laat zien hoe een team wordt gecreëerd binnen een applicatie.



We beginnen met een beslismoment waar we controleren of de gebruiker is ingelogd in de applicatie. Als de gebruiker niet ingelogd is, laten we hem of haar inloggen. Is de gebruiker wel ingelogd, dan tonen we het hoofdscherm van de applicatie. Vanaf het hoofdscherm heeft de gebruiker de optie om naar "Beheer je teams" te gaan, waar hij of zij een teamnaam en teambeschrijving kan instellen. Vervolgens controleren we of de gekozen teamnaam al bestaat binnen het systeem. Als de naam al bestaat, wordt de gebruiker geïnformeerd dat het team niet gemaakt kan worden en wordt gevraagd een andere naam te kiezen. Als de naam uniek is, wordt het team succesvol gecreëerd en wordt dit bevestigd aan de gebruiker, waarna de flow terugkeert naar het hoofdscherm. Hier eindigt het proces.

Event Creation Diagram: Laat zien hoe een evenement wordt aangemaakt binnen een applicatie.



We beginnen met een beslismoment waar we controleren of de gebruiker is ingelogd in de applicatie. Als de gebruiker niet ingelogd is, wordt hij of zij naar het inlogscherm geleid, wat essentieel is voor het beveiligen van toegang tot de functies die betrekking hebben op het beheren van evenementen.

Na het inloggen wordt de gebruiker naar het hoofdscherm van de app geleid. Hier kan de gebruiker kiezen voor de optie "Beheer Evenementen". Dit is de gateway voor het creëren en beheren van evenementen, wat centraal staat in de app-functionaliteit.

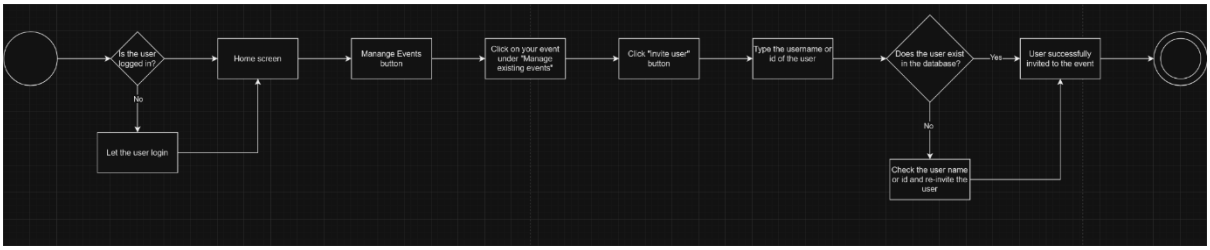
In het "Beheer Evenementen"-scherm kan de gebruiker een nieuw evenement aanmaken door het invullen van een formulier. Hier worden essentiële details gevraagd zoals de naam van het evenement, locatie, tijd, en andere relevante informatie. Vervolgens wordt gecontroleerd of alle ingevoerde gegevens correct zijn ingevuld.

Als de gegevens niet correct zijn, wordt de gebruiker gevraagd deze aan te passen. Dit zorgt ervoor dat de informatie van evenementen nauwkeurig en nuttig is.

Als alle inputs correct zijn, wordt het evenement succesvol aangemaakt. De gebruiker krijgt een bevestiging van de succesvolle creatie, wat zorgt voor een duidelijke communicatie en afsluiting van het proces.

Dit proces verzekert een georganiseerde en effectieve manier om evenementen binnen de applicatie te beheren, met een sterke focus op gebruikersverificatie en data-integriteit.

User Invitation to Event Diagram: Laat zien hoe een gebruiker wordt uitgenodigd voor een evenement binnen de applicatie.



We beginnen met een beslismoment waar we controleren of de gebruiker is ingelogd in de applicatie. Indien de gebruiker niet ingelogd is, wordt deze naar het inlogscherm geleid om in te loggen, wat de toegangscontrole verzekert.

Na succesvolle login komt de gebruiker terecht op het hoofdscherm van de app. Van daaruit kan de gebruiker de "Beheer Evenementen" knop selecteren, wat hem naar het overzicht van beschikbare evenementen leidt.

In het overzichtsscherm selecteert de gebruiker een specifiek evenement en klikt vervolgens op de knop "Gebruiker uitnodigen". Hier wordt de gebruiker gevraagd om de gebruikersnaam of ID in te voeren van de persoon die hij wil uitnodigen.

Na het invoeren van de gebruikersnaam of ID, controleert het systeem of deze gebruiker bestaat in de database. Als de gebruiker bestaat, wordt de uitnodiging succesvol verzonden en is het proces voltooid.

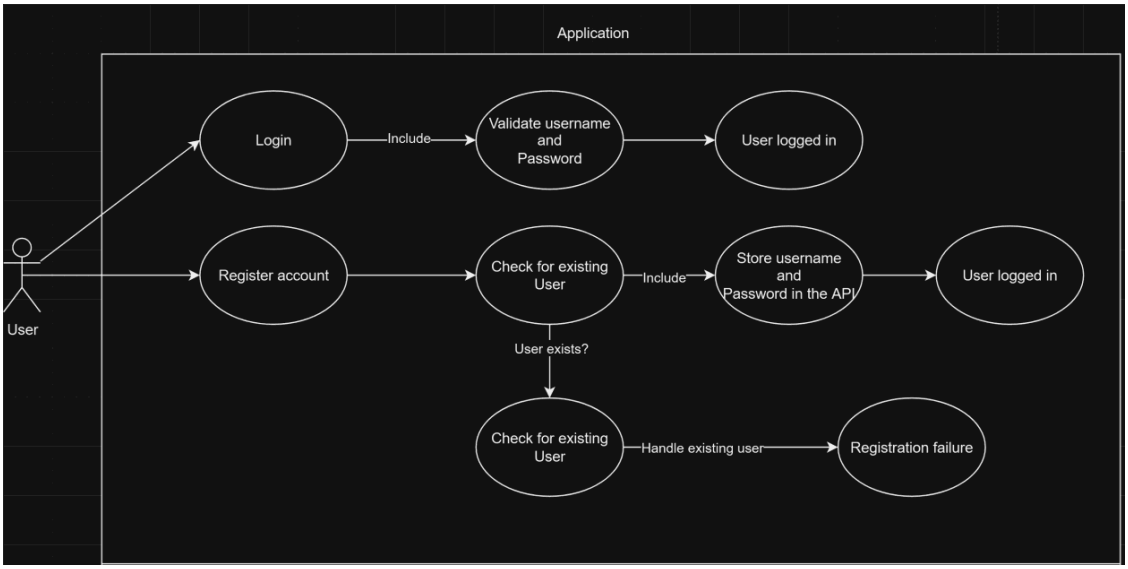
Indien de gebruiker niet gevonden wordt, krijgt de uitnodiger een melding om de invoer te controleren en opnieuw te proberen. Dit helpt fouten te voorkomen en zorgt ervoor dat alleen bestaande gebruikers worden uitgenodigd.

Dit proces zorgt voor een gestructureerde en veilige manier om gebruikers uit te nodigen voor evenementen, waarbij de integriteit van de gebruikersgegevens gewaarborgd blijft en alleen bevoegde gebruikers acties kunnen uitvoeren.

3.2 Use Case Diagram

Use Case diagram: Laat zien hoe verschillende gebruikers (actoren) interacties hebben met het systeem.

User Authentication Use Case Diagram:



We beginnen met een controlepunt waar we kijken of de gebruiker is ingelogd in de applicatie. Indien de gebruiker niet is ingelogd, wordt hij of zij naar het inlogscherm gestuurd. Dit is een cruciale stap voor de veiligheid, aangezien de toegang verleent tot functies die gerelateerd zijn aan gebruikersbeheer en authenticatie.

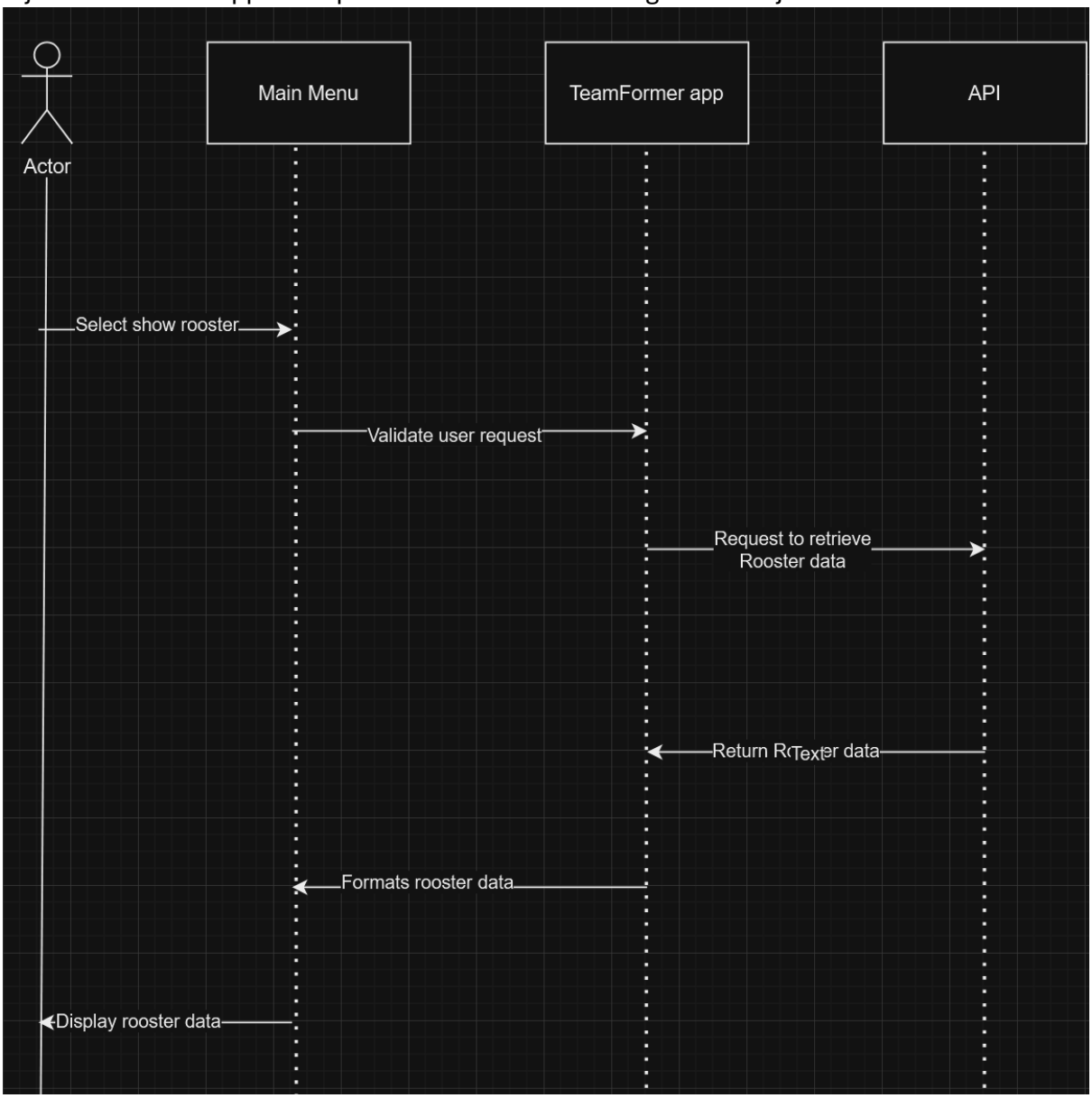
Eenmaal ingelogd, krijgt de gebruiker toegang tot het hoofdscherm van de applicatie. Vanuit dit scherm kan de gebruiker kiezen om een account te registreren of direct in te loggen, afhankelijk van de behoefte. Bij het registreren van een account moet de gebruiker belangrijke informatie verstrekken zoals gebruikersnaam en wachtwoord, welke vervolgens gevalideerd worden door het systeem.

Tijdens de validatie wordt gecheckt of de gebruikersnaam al bestaat middels de "Check for Existing User". Indien de gebruiker al bestaat, wordt het registratieproces onderbroken en krijgt de gebruiker een melding van falen. Als de gebruikersnaam uniek is, worden de gegevens opgeslagen in de API, en wordt de registratie als voltooid beschouwd, waarna de gebruiker wordt ingelogd in het systeem.

Deze stappen zorgen voor een gedegen en veilig proces rondom gebruikersauthenticatie, waarbij zowel de integriteit van de gebruikersgegevens als de toegangsbeveiliging tot de applicatie centraal staan. Dit proces benadrukt het belang van nauwkeurige data-invoer en het correct afhandelen van authenticatieprocedures binnen de applicatie.

3.3 Use Case Diagram UML Sequence Diagram

Sequence diagram: Beschrijft de interactie tussen objecten in een sequentiele volgorde. Bijvoorbeeld: de stappen die plaatsvinden wanneer een gebruiker zijn rooster wilt zien.



We beginnen met een interactie waarin de gebruiker de optie 'Toon rooster' selecteert vanuit het hoofdmenu van de applicatie. Deze stap is cruciaal om de gebruiker toegang te geven tot zijn of haar gepersonaliseerde roostergegevens, die centraal staan in de functionaliteit van de app.

Na de selectie valideert de TeamFormer app de aanvraag van de gebruiker. Deze validatie is essentieel om te verzekeren dat de verzoeken legitiem zijn en dat de data alleen toegankelijk is voor geautoriseerde gebruikers, wat de veiligheid binnen de applicatie verhoogt.

Zodra de aanvraag is gevalideerd, stuurt de app een verzoek naar de API om de roostergegevens op te halen. Dit toont de integratie van de app met externe databronnen, waardoor de app dynamisch en up-to-date informatie kan leveren.

De API verwerkt het verzoek en stuurt de roostergegevens terug naar de app. Dit punt illustreert de efficiëntie en de snelheid van de netwerkcommunicatie binnen de applicatie-architectuur.

Bij ontvangst formatteert de TeamFormer app de data voor weergave. Dit formateringsproces is cruciaal om de gegevens leesbaar en gemakkelijk interpreteerbaar te maken voor de gebruiker, wat bijdraagt aan een betere gebruikerservaring.

Ten slotte wordt het rooster weergegeven aan de gebruiker. Waarbij de gebruiker een helder en nauwkeurig beeld krijgt van zijn of haar geplande activiteiten. Dit zorgt voor duidelijke communicatie en een effectieve afsluiting van het proces.

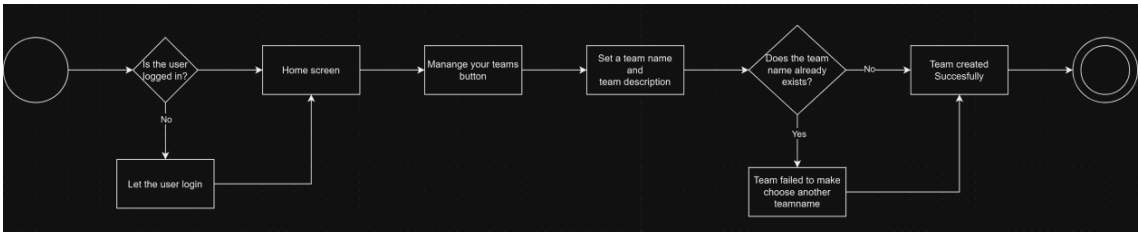
Dit proces verzekert een gestructureerde en effectieve manier om roostergegevens binnen de TeamFormer app te beheren, met een sterke focus op gebruikersverificatie en data-integriteit.

3.4 Advies

UML advies:

Tijdens de ontwikkeling van onze applicatie hebben wij meerde UML diagrammen opgesteld voor de Crossplatform app. Deze diagrammen visualiseert het processen vanaf het inloggen van de gebruiker tot het succesvol creëren van een team en meer. Hieronder bieden wij een overzicht van deze processen en verklaren wij waarom deze specifieke aanpak gekozen is.

Team Management Diagram:



Waarom Deze Werkproces?

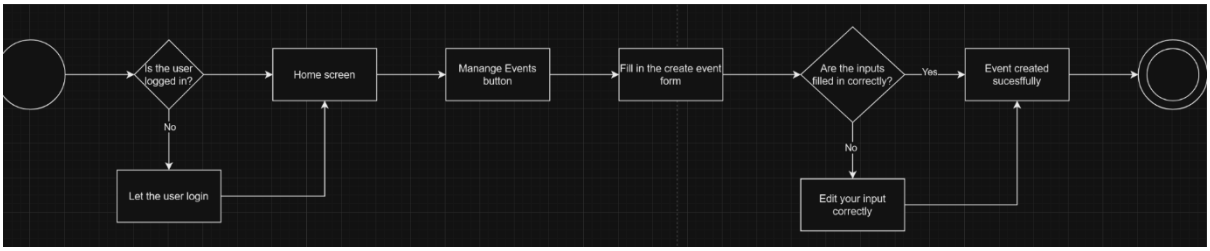
Dit werkproces is gekozen vanwege meerdere redenen:

1. **Veiligheid en Autorisatie**: Door alleen ingelogde gebruikers toe te staan teams te beheren, wordt de veiligheid en integriteit van de app gewaarborgd.
2. **Gebruiksgemak**: De stroomlijning van het proces van het hoofdscherm naar de teamcreatie maakt het voor gebruikers intuïtief en eenvoudig om te navigeren en nieuwe teams te vormen.
3. **Uniciteit van Teams**: Het controleren van de teamnaam op uniciteit voorkomt verwarring en fouten in teambeheer, wat essentieel is voor een heldere communicatie en organisatie binnen de app.
4. **Feedback en Bevestiging**: Duidelijke feedback en bevestiging aan de gebruiker bij elke stap zorgt voor een betrouwbare en vriendelijke gebruikerservaring.

Conclusie

Het ontworpen teammanagementproces biedt een solide basis voor het efficiënt en veilig beheren van teams binnen onze applicatie. Dit proces verhoogt niet alleen de tevredenheid en het vertrouwen van de gebruikers maar bevordert ook de operationele efficiëntie binnen de app. We adviseren om dit proces te implementeren zoals beschreven in het activiteitendiagram en continue feedback van gebruikers te verzamelen om het verder te optimaliseren.

Event Creation Diagram: Laat zien hoe een evenement wordt aangemaakt binnen een applicatie.



Waarom Dit Werkproces?

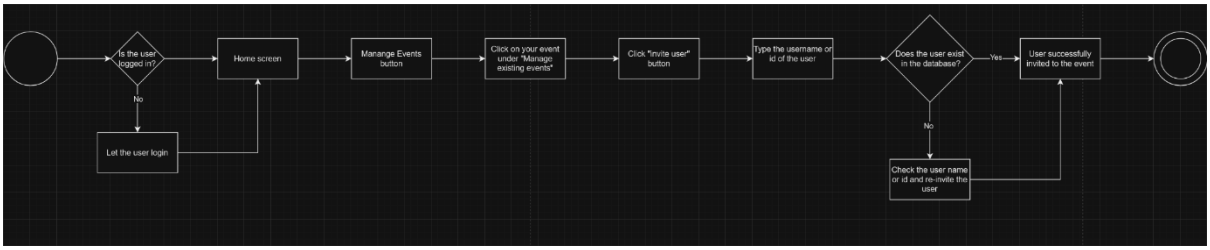
Dit werkproces is gekozen vanwege meerdere redenen:

1. **Veiligheid en Autorisatie**: Door alleen ingelogde gebruikers toe te staan evenementen aan te maken, wordt de veiligheid en integriteit van de app gewaarborgd. Dit zorgt ervoor dat alleen geautoriseerde gebruikers evenementen kunnen beheren, wat belangrijk is voor de controle en beveiliging van de applicatie.
2. **Gebruiksgemak**: De stroomlijning van het proces van het hoofdscherm naar het aanmaken van een evenement maakt het voor gebruikers intuïtief en eenvoudig om te navigeren en nieuwe evenementen te organiseren. Dit verhoogt de gebruikerstevredenheid en vermindert de kans op fouten tijdens het invulproces.
3. **Data-integriteit**: Door de validatie van de ingevoerde gegevens te controleren, wordt verzekerd dat de informatie van evenementen nauwkeurig is. Dit voorkomt problemen zoals dubbele evenementen of incorrecte gegevens, wat essentieel is voor een goede organisatie en communicatie binnen de app.
4. **Feedback en Bevestiging**: Het bieden van duidelijke feedback en bevestiging aan de gebruiker bij elke stap van het evenementcreatieproces zorgt voor een betrouwbare en vriendelijke gebruikerservaring. Dit helpt bij het opbouwen van vertrouwen en zorgt ervoor dat gebruikers duidelijk begrijpen wat er gebeurt met hun acties binnen de applicatie.

Conclusie

Het ontworpen proces voor het aanmaken van evenementen biedt een solide basis voor het efficiënt en veilig beheren van evenementen binnen onze applicatie. Dit proces verhoogt niet alleen de tevredenheid en het vertrouwen van de gebruikers maar bevordert ook de operationele efficiëntie binnen de app. We adviseren om dit proces te implementeren zoals beschreven in het activiteitendiagram en continue feedback van gebruikers te verzamelen om het verder te optimaliseren.

User Invitation to Event Diagram: Laat zien hoe een gebruiker wordt uitgenodigd voor een evenement binnen de applicatie.



Waarom Deze Werkproces?

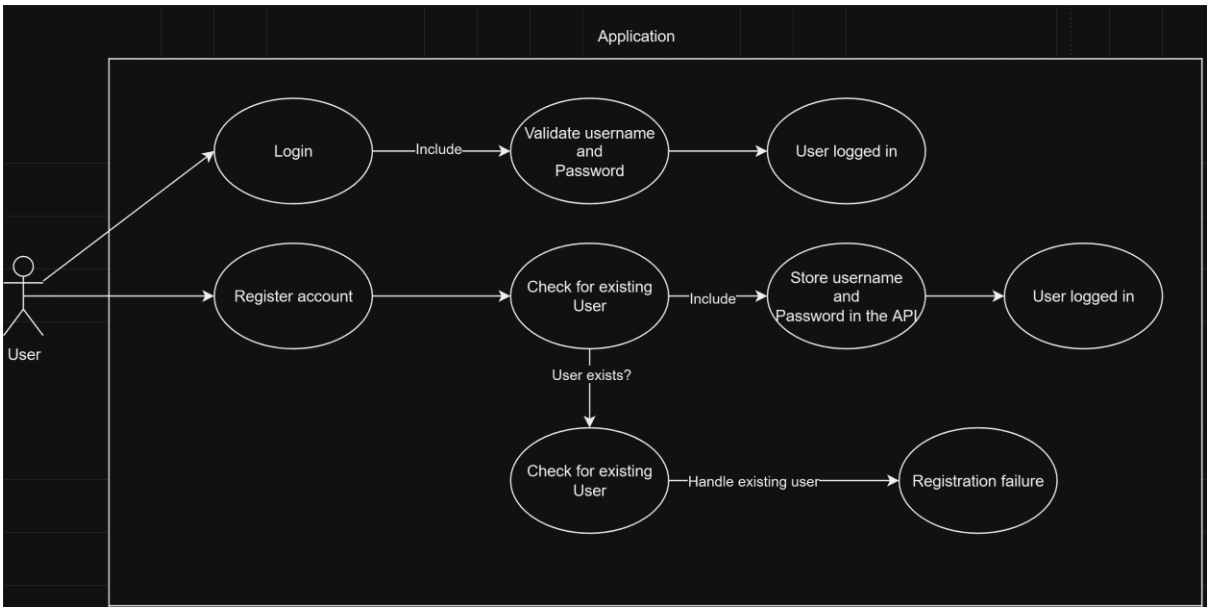
Dit werkproces is gekozen vanwege meerdere redenen:

1. **Veiligheid en Autorisatie**: Door alleen ingelogde gebruikers toe te staan evenementen uit te nodigen, wordt de veiligheid en integriteit van de app gewaarborgd. Dit zorgt ervoor dat alleen geautoriseerde gebruikers toegang hebben tot functies die impact kunnen hebben op de privacy en de ervaring van andere gebruikers.
2. **Gebruiksgemak**: De stroomlijning van het proces van het hoofdscherm naar de uitnodiging van gebruikers maakt het voor gebruikers intuïtief en eenvoudig om te navigeren en nieuwe uitnodigingen te versturen. Dit verhoogt de efficiëntie van gebruikerinteracties binnen de app.
3. **Validatie van Gebruikersgegevens**: Door te controleren of de gebruiker bestaat in de database voordat een uitnodiging wordt verstuurd, wordt voorkomen dat uitnodigingen naar niet-bestaande gebruikers worden gestuurd. Dit vermindert de kans op fouten en zorgt voor een gerichtere en relevantere gebruikersinteractie.
4. **Feedback en Bevestiging**: Duidelijke feedback en bevestiging aan de gebruiker bij elke stap van het uitnodigingsproces zorgen voor een betrouwbare en vriendelijke gebruikerservaring. Gebruikers worden adequaat geïnformeerd over de status van hun acties, wat bijdraagt aan een positieve interactie met de app.

Conclusie

Het ontworpen proces voor het uitnodigen van gebruikers voor evenementen biedt een solide basis voor het efficiënt en veilig beheren van gebruikersinteracties binnen onze applicatie. Dit proces verhoogt niet alleen de tevredenheid en het vertrouwen van de gebruikers maar bevordert ook de operationele efficiëntie binnen de app. We adviseren om dit proces te implementeren zoals beschreven in het activiteitendiagram en continue feedback van gebruikers te verzamelen om het verder te optimaliseren.

User Authentication Use Case Diagram:



Waarom Dit Werkproces?

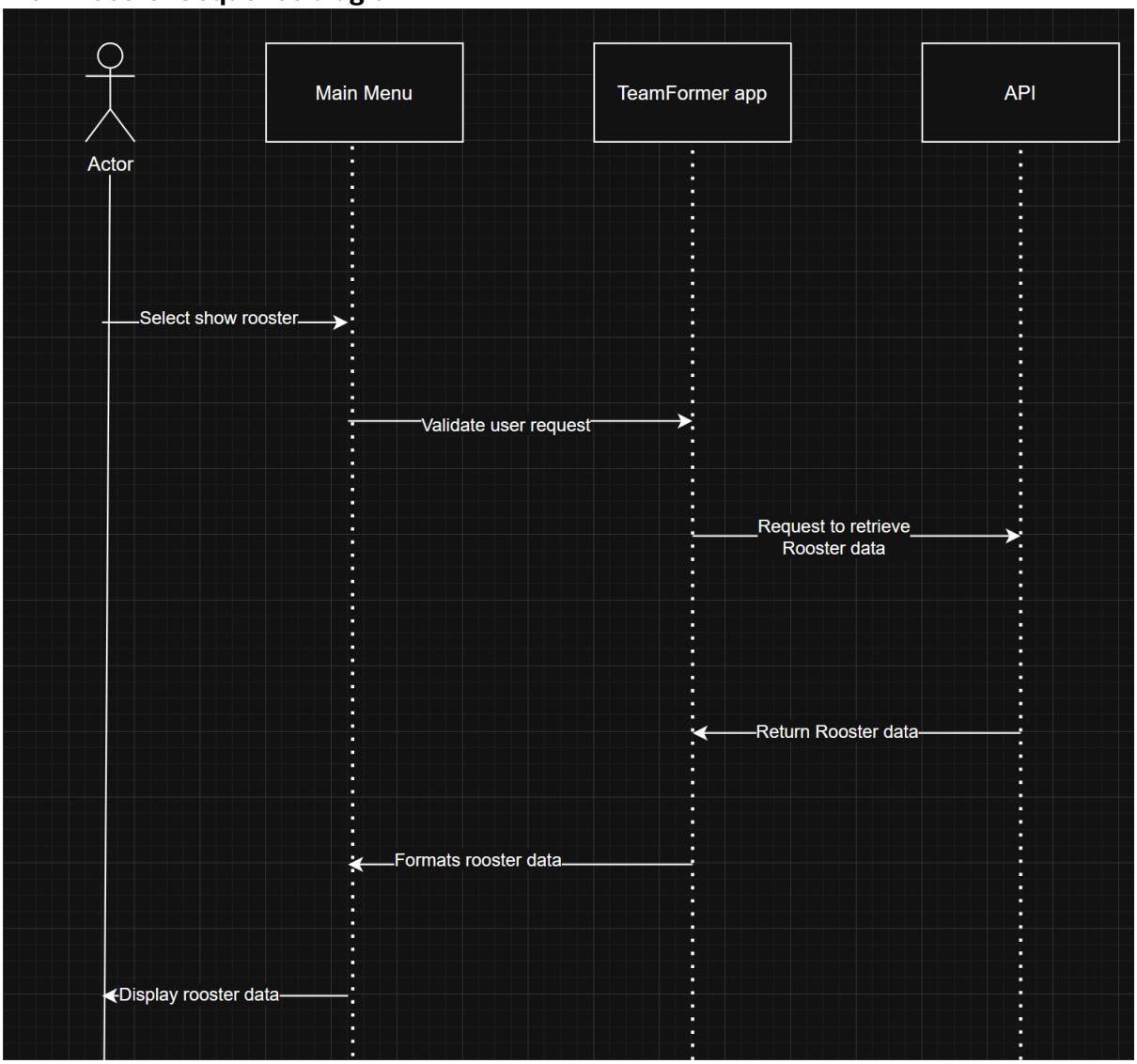
Dit werkproces is gekozen vanwege meerdere redenen:

1. **Veiligheid en Autorisatie**: Door alleen ingelogde of correct geregistreerde gebruikers toegang te geven tot belangrijke functies, wordt de veiligheid en integriteit van de app gewaarborgd. Dit is cruciaal voor het beschermen van gebruikersgegevens en het voorkomen van ongeautoriseerde toegang.
2. **Gebruiksgemak**: De stroomlijning van het proces van het hoofdscherm naar het login- en registratiescherm maakt het voor gebruikers intuïtief en eenvoudig om toegang te krijgen tot hun accounts of nieuwe accounts te creëren. Dit vermindert de kans op gebruikersfrustratie en verbetert de algemene gebruikerservaring.
3. **Validatie van Gegevens**: Het controleren van de gegevens van de gebruikersnaam voorkomt duplicatie en fouten bij het accountbeheer. Dit is essentieel voor een duidelijke communicatie en organisatie binnen de app, omdat het zorgt dat elke gebruiker een unieke identiteit heeft.
4. **Feedback en Bevestiging**: Duidelijke feedback en bevestiging aan de gebruiker bij elke stap van het inlog- en registratieproces zorgen voor een betrouwbare en vriendelijke gebruikerservaring. Dit helpt gebruikers te begrijpen dat hun acties succesvol zijn uitgevoerd of dat correcties nodig zijn.

Conclusie

Het ontworpen authenticatieproces biedt een solide basis voor het efficiënt en veilig beheren van gebruikerstoegang binnen onze applicatie. Dit proces verhoogt niet alleen de tevredenheid en het vertrouwen van de gebruikers, maar bevordert ook de operationele efficiëntie binnen de app. We adviseren om dit proces te implementeren zoals beschreven in het activiteitendiagram en continue feedback van gebruikers te verzamelen om het verder te optimaliseren

View Rooster Sequence diagram:



Waarom Dit Werkproces?

Dit werkproces is gekozen vanwege meerdere redenen:

1. **Veiligheid en Autorisatie**: Door het proces te beginnen met een validatie van de gebruikersaanvraag, wordt gewaarborgd dat alleen geautoriseerde gebruikers toegang hebben tot roostergegevens. Dit is essentieel voor het beschermen van persoonlijke en gevoelige informatie binnen de app.
2. **Efficiëntie in Gegevensopvraging**: Door de interactie met een externe API voor het ophalen van de roosterdata te automatiseren, wordt een snelle en betrouwbare gegevensuitwisseling gegarandeerd. Dit verhoogt de efficiëntie van het systeem en vermindert de wachttijd voor de gebruiker.

3. **Gebruiksgemak**: het proces van de aanvraag tot de weergave van het rooster, maakt het voor gebruikers intuïtief en eenvoudig om hun roosterinformatie te bekijken. Dit verbetert de algehele gebruikerservaring door de noodzakelijke informatie gemakkelijk toegankelijk te maken.
4. **Feedback en Bevestiging**: Het formatteerproces van de roosterdata en de uiteindelijke weergave zorgen voor duidelijke en begrijpelijke feedback aan de gebruiker. Elke stap zorgt voor betrouwbare informatieoverdracht, waardoor gebruikers vertrouwen hebben in de nauwkeurigheid van de gegevens.

Conclusie

Het ontworpen proces van de roosterinformatie biedt een solide basis voor een efficiënte en veilige gegevenstoegang binnen de TeamFormer app. Dit proces verhoogt niet alleen de tevredenheid van de gebruikers door snelle toegang tot relevante informatie maar bevordert ook de operationele efficiëntie binnen de app. We adviseren om dit proces te implementeren zoals beschreven in het sequentiediagram en continue feedback van gebruikers te verzamelen om het verder te optimaliseren.

3.5 Onbrekende requirements & advies

Bij de ontwikkeling van de TeamFormer app met Flutter en Dart zijn twee belangrijke functionaliteiten niet gerealiseerd: een agenda voor evenementen en een QR-code uitnodigingssysteem voor teamleden. Deze functionaliteiten zijn essentieel om de gebruikerservaring te verbeteren en te voldoen aan de behoeften van de gebruikers.

1. Agenda voor Evenementen Analyse: Het ontbreken van een agendafunctionaliteit beperkt de mogelijkheid voor gebruikers om evenementen effectief te plannen en te beheren binnen de app.

Advies:
Toevoeging van de Agenda Functionaliteit: Implementeer een agenda-module die gebruikers in staat stelt evenementen aan te maken, te bekijken en te beheren. Gebruik Flutter packages zoals **table_calendar** of **flutter_calendar_carousel** die een uitgebreide ondersteuning bieden voor kalenderweergaven en interacties. Integratie met Externe Kalenders: Overweeg integratie met Google Calendar of Outlook om synchronisatie met externe kalenders mogelijk te maken, waardoor gebruikers hun persoonlijke en professionele planning kunnen stroomlijnen.

2. QR-code Uitnodigingssysteem voor Teamleden Analyse:
   Het gebrek aan een QR-code uitnodigingssysteem maakt het moeilijk voor gebruikers om snel en gemakkelijk nieuwe leden aan teams toe te voegen.

Advies:
Implementatie van QR-code Functionaliteit: Gebruik Flutter packages zoals **qr_flutter** en **barcode_scan** om QR-codes te genereren en te scannen. Hiermee kunnen teamleiders snel een QR-code genereren die teamleden kunnen scannen om zich bij een team aan te sluiten. Gebruiksvriendelijke Interface: Zorg ervoor dat de interface voor het genereren en scannen van QR-codes intuïtief en toegankelijk is. Implementeer stapsgewijze instructies binnen de app om gebruikers te begeleiden in het proces.

Conclusie:
De toevoeging van een geavanceerde agenda voor evenementen en een QR-code uitnodigingssysteem zijn cruciale stappen naar het vervullen van de gebruikersbehoeften binnen de TeamFormer app. Deze aanpassingen zullen niet alleen de functionaliteit van de app verbeteren, maar ook zorgen voor een hogere gebruikerstevredenheid en een beter engagement met de app. Het wordt aanbevolen om deze veranderingen systematisch te implementeren, te beginnen met een grondige testing fase om de integratie met de bestaande app-structuur te verzekeren.

4. Testplan

4.1 Doel van het Testplan

Het doel van dit testplan is om de kwaliteit van de Flutter-applicatie te garanderen door middel van geautomatiseerde en handmatige tests die de verschillende functionaliteiten valideren. Dit document richt zich op het testen van het teambeheer en evenementenbeheer binnen de applicatie.

De applicatie bevat meerdere functionaliteiten, waaronder:

- Teams aanmaken, ophalen en beheren.
- Evenementen aanmaken, ophalen en beheren.
- Gebruikersauthenticatie.

4.2 Testomgeving

De geautomatiseerde tests worden uitgevoerd via een Continuous Integration (CI) workflow in GitHub Actions. De configuratie van de testomgeving is als volgt:

- **Operating System:** Ubuntu-latest
- **Flutter versie:** 3.27.1
- **Dart versie:** ^3.5.1

4.3 Testscope

De scope van de tests omvat:

- CRUD-functionaliteiten voor teams en evenementen.
- Navigatie tussen pagina's en interacties met gebruikersinterface-elementen.
- Integriteit van de applicatie tijdens het bouwen en uitvoeren.

4.4 Teststrategie

Geautomatiseerde tests

De volgende stappen worden uitgevoerd binnen de GitHub Actions workflows:

Flutter CI Workflow:

1. **Code ophalen:** De repository wordt opgehaald uit GitHub.
2. **Flutter instellen:** Flutter versie 3.27.1 wordt geïnstalleerd.
3. **Dependencies installeren:** Alle vereiste pakketten worden opgehaald via flutter pub get.
4. **Tests uitvoeren:** De test suite wordt uitgevoerd met flutter test.

Cross Platform Development - Team 2

Flutter CI badge indicates the status of the continuous integration (CI) pipeline for the project. The CI pipeline runs automated tests on every pull request and push to the `main` branch to ensure code quality and catch potential issues before merging.

Web Release Workflow:

1. **Code ophalen:** De repository wordt opgehaald.
2. **Flutter instellen:** Flutter versie 3.27.1 wordt geïnstalleerd.
3. **Dependencies installeren:** Alle vereiste pakketten worden opgehaald.
4. **Webversie bouwen:** De applicatie wordt gebouwd voor webgebruik met flutter build web.
5. **Build controleren:** De output wordt gecontroleerd in de map build/web.

Build Web Release badge represents the status of the **Build Web Release** GitHub Actions workflow for this project. The workflow is triggered automatically whenever code is pushed to the `main` branch. Its purpose is to ensure that the Flutter application can be successfully built for web deployment.

- **Green Badge**: The workflow successfully completed all steps, including dependency installation and building the web version of the app. This confirms that the code in the `main` branch is ready for web deployment.
- **Red Badge**: The workflow encountered an error during one of its steps, such as dependency installation or the web build process. This indicates that the code needs to be fixed before it can be deployed for web.

4.5 Integratietest in de Flutter-applicatie

Wat doet de integratietest?

De integratietest valideert de volledige gebruikersstroom voor het inloggen binnen de applicatie. Het test de volgende functionaliteiten:

1. Start de applicatie: Controleert of de app correct wordt geladen.
2. Vult gebruikersinvoer in:

- Gebruikersnaam (username_field): "Renas".
- Wachtwoord (password_field): "Renas123".

3. Voert een actie uit:

- Drukt op de knop "Login" (login_button).

4. Controleert de navigatie:

- Verifieert of de gebruiker succesvol wordt doorgestuurd naar de HomePage met de tekst: "Welcome, Renas!".

De test simuleert realistische gebruikersinteracties en valideert of de applicatie correct reageert op invoer en acties.

Hoe kan de intergratie test uitvoeren?

Gebruik het volgende commando in de terminal om de integratietest uit te voeren:

flutter drive --driver=test_driver/integration_test_driver.dart --target=integration_test/login_integration_test.dart

4.6 Handmatige tests

Naast de geautomatiseerde tests voeren we de volgende handmatige tests uit:

Teams

1. **Teams aanmaken:**

Controleer of een nieuw team succesvol kan worden aangemaakt.

Valideer dat de ingevoerde gegevens correct worden opgeslagen in de backend.

Test foutscenario's, zoals lege invoervelden of ongeldige gegevens.

2. **Teams ophalen (fetchen):**

Controleer of de lijst met teams correct wordt opgehaald en weergegeven.

Test de weergave van dynamische updates, zoals het toevoegen of verwijderen van een team.

Evenementen

1. **Evenementen aanmaken:**

Controleer of een nieuw evenement succesvol kan worden aangemaakt.

Test scenario's met ontbrekende of ongeldige invoerwaarden.

2. **Evenementen ophalen (fetchen):**

Controleer of de lijst met evenementen correct wordt opgehaald en weergegeven.

Test op filtering en sortering van evenementen.

3. **Evenementen beheren:**

Test het bewerken en verwijderen van evenementen.

Stappen bij handmatig testen

1. Navigeren door de applicatie: Controleer of knoppen en links correct functioneren.
2. Gegevens invoeren: Test met zowel correcte als incorrecte invoer om randgevallen en foutafhandeling te valideren.
3. Output controleren: Valideer dat gegevens correct worden weergegeven in de UI en opgeslagen in de database.

4.7 Testdoelen

Algemene Doelen

1. Verifiëren dat gebruikers succesvol teams en evenementen kunnen beheren.
2. Valideren van de gebruikerservaring door handmatige tests.

Specifieke Testdoelen

1. **CRUD-functionaliteiten testen:**

Controleer of teams correct kunnen worden aangemaakt, bewerkt, verwijderd en weergegeven.

Valideer vergelijkbare functionaliteiten voor evenementen.

2. **Integriteit van de build:**

Zorg ervoor dat de web- en mobiele builds zonder fouten worden uitgevoerd.

3. **Navigatie en interactie:**

Controleer of de applicatie intuïtief is en gebruikers eenvoudig door de verschillende secties kunnen navigeren.

Succescriteria

1. Alle CRUD-operaties moeten succesvol worden uitgevoerd zonder fouten.
2. De applicatie moet foutloos bouwen en functioneren in zowel de testomgeving als de productieomgeving.
3. Alle handmatige tests moeten zonder kritieke problemen worden doorstaan.
4. Feedback van handmatige tests moet direct worden verwerkt om verbeteringen door te voeren.

5. (Sprint) Planning

Sprintduur: Elke sprint duurt 2 weken.

Team: 2 developers werken aan de sprint.

Doel van Sprint Planning

Vaststellen welke user stories in de sprint worden opgenomen.

Focus op het leveren van waardevolle functionaliteiten.

Voorbereiding

Backlog: Het team zorgt voor een geprioriteerde en verfijnde backlog met duidelijke user stories.

Selectie van User Stories

Kies user stories die haalbaar zijn binnen de sprintduur.

Inschatting van de benodigde inspanning per story.

Taken Opstellen

User stories worden opgesplitst in kleinere, beheersbare taken.

Toewijzing van taken aan developers op basis van expertise.

Risico's en Obstakels

Identificeer en bespreek mogelijke risico's of obstakels voor de sprint.

Definition of Done (DoD)

Stel criteria vast voor wat het betekent dat een taak als "klaar" wordt beschouwd.

Commitment

Het team committeert zich aan de geselecteerde user stories voor de sprint.

6. Sprint Backlog

Definitie

De Sprint Backlog is een lijst van user stories en taken die het team zich heeft gecommitteerd om binnen de huidige sprint te voltooien.

Inhoud van de Sprint Backlog

Geselecteerde User Stories: User stories die zijn gekozen voor de sprint, met duidelijke acceptatiecriteria.

Taken: Gedetailleerde taken die zijn afgeleid van de user stories, opgesplitst in beheersbare eenheden.

Schattingspunten: Inschatting van de inspanning voor elke user story en taak, vaak in story points of uren.

Prioriteiten: Taken zijn geprioriteerd op basis van waarde en afhankelijkheden.

Beheer

Aanpassingen: De Sprint Backlog kan tijdens de sprint worden aangepast indien nodig, maar dit moet zorgvuldig worden gedaan om de focus en doelstellingen van de sprint te waarborgen.

Dagelijkse Updates: Het team bespreekt de voortgang van de Sprint Backlog tijdens dagelijkse stand-ups, waarbij taken worden bijgewerkt op basis van de voortgang en eventuele obstakels.

Doelstelling

De Sprint Backlog is bedoeld om het team te helpen gefocust te blijven op de doelen van de sprint en om transparantie te bieden over de voortgang naar de stakeholders.