

# Software Design Document



Door: Renas Khalil en Sadek al Mousawi

Groep: Groep 2 RS

Datum: 26 September 2024

Versie: 1.1

Vak: Cross-Platform Development (24/25)

## Inhoudsopgave

1. Requirements .....	4
1.1 Functionele eisen.....	4
1.2 Niet-functionele eisen .....	6
2. Functioneel Ontwerp.....	7
2.1 Doel van het Systeem .....	7
2.2 Functionele Structuur van het Systeem .....	7
2.3 Volledigheid van Requirements .....	8
2.4 Controle op Gebruikerswensen .....	8
2.5 Wireframes .....	9
2.6 Activity Diagram .....	9
2.7 Advies .....	10
3. Technisch Ontwerp .....	11
3.1 Externe koppelingen .....	11
3.2 Datamodels .....	11
3.3 Architectuur .....	11
3.4 Interacties .....	11
4. Testplan .....	12
4.1 Teststrategie .....	12
Testdoelen.....	12
Wat moet wel (en niet) worden getest?.....	12
Welke soorten testen? .....	12
Welke tools zijn hiervoor nodig? .....	12
4.2 Acceptatie testen .....	12
Test cases.....	12
4.3 Unit-testen .....	12
4.4 Widget-testen .....	12
4.5 Integratie-testen .....	12
5. Goals & Milestones .....	13

6.	Global timeline .....	14
7.	Definition of Done .....	15
7.1	Voor de developer: .....	15
7.2	Voor de gebruiker: .....	15
7.3	Algemeen voor het team: .....	16
8.	(Sprint) Planning .....	17
8.1	Doel van Sprint Planning .....	17
8.2	Vorbereiding .....	17
8.3	Selectie van User Stories .....	17
8.4	Taken Opstellen .....	17
8.5	Risico's en Obstakels .....	17
8.6	Definition of Done (DoD) .....	17
8.7	Commitment .....	17
9.	Sprint Backlog.....	18
9.1	Definitie .....	18
9.2	Inhoud van de Sprint Backlog .....	18
9.3	Beheer .....	18
9.4	Doelstelling .....	18

# 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.

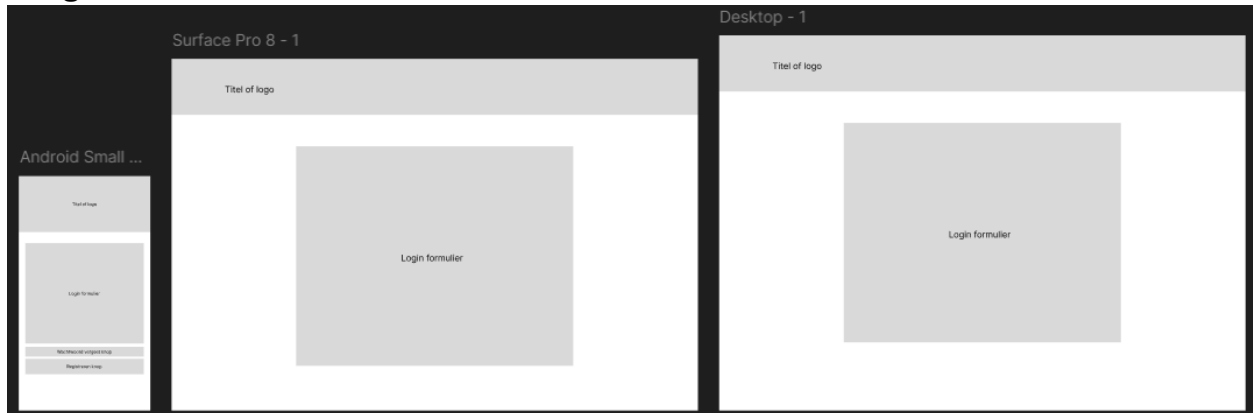
**Flexibiliteit:** Het systeem moet genoeg flexibiliteit bieden voor verschillende soorten teams en activiteiten. Denk aan meerdere rollen, de mogelijkheid om teams aan te passen, en het delen van informatie via links of QR-codes.

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



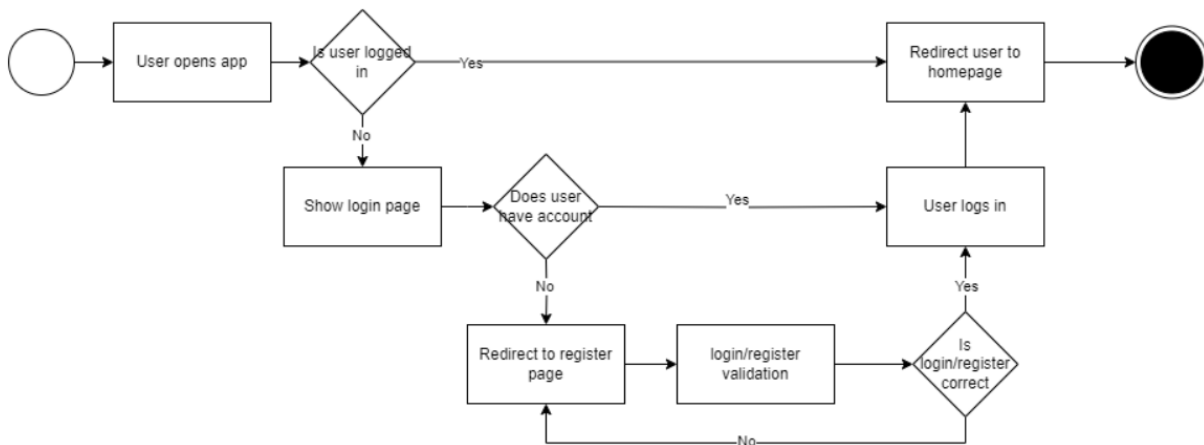
## 2.5 Wireframes

### Inlogscherf Wireframe



Dit is het wireframe van het inlogscherf. Gebruikers moeten verplicht inloggen om de applicatie te kunnen gebruiken. Bij het openen van de applicatie wordt het inlogscherf weergegeven. Als een gebruiker al is ingelogd, wordt hij of zij automatisch doorgestuurd naar de homepagina.

## 2.6 Activity Diagram



1. User opens app: De gebruiker opent de applicatie.

2. Is user logged in?: Het systeem controleert of de gebruiker al is ingelogd.

Ja: De gebruiker wordt direct doorgestuurd naar de homepage.

Nee: Het systeem toont de loginpagina.

3.Does user have an account?: De gebruiker moet aangeven of hij al een account heeft.

Ja: De gebruiker voert zijn inloggegevens in.

Nee: De gebruiker wordt doorgestuurd naar de registratiepagina om een account aan te maken.

4.Login/Register validation: Het systeem controleert de ingevoerde gegevens (bij zowel inloggen als registreren).

Correct: De gebruiker wordt ingelogd.

Niet correct: De gebruiker wordt teruggestuurd naar de inlog- of registratieschermen om het opnieuw te proberen.

5.User logs in: Als de gegevens correct zijn, wordt de gebruiker ingelogd.

6.Redirect user to homepage: Na succesvol inloggen wordt de gebruiker naar de homepage doorgestuurd.

## 2.7 Advies

Als uit deze analyse blijkt dat er bepaalde tekortkomingen zijn, zoals ontbrekende functionaliteiten of gebruikerswensen die niet volledig vervuld kunnen worden, zal er een advies worden gegeven. Dit advies kan bestaan uit:

**Toevoeging van ontbrekende requirements:** Voorstellen voor ontbrekende functionaliteiten die nodig zijn om aan de gebruikerswensen te voldoen.

**Aanpassingen aan de huidige functionaliteiten:** Suggesties voor het verbeteren van de huidige functies om beter aan de behoeften van de gebruikers te voldoen.

## 3. Technisch Ontwerp

### 3.1 Externe koppelingen

### 3.2 Datamodels

### 3.3 Architectuur

### 3.4 Interacties

## 4. Testplan

### 4.1 Teststrategie

Testdoelen

Wat moet wel (en niet) worden getest?

Welke soorten testen?

Welke tools zijn hiervoor nodig?

### 4.2 Acceptatie testen

Test cases

### 4.3 Unit-testen

### 4.4 Widget-testen

### 4.5 Integratie-testen

## 5. Goals & Milestones

## 6. Global timeline

## 7. Definition of Done

### 7.1 Voor de developer:

Een taak of functionaliteit wordt als "klaar" beschouwd door de developer wanneer:

Code is geschreven en voldoet aan de vereisten van de user story: De functionaliteit moet precies werken zoals beschreven in de user story en voldoen aan de acceptatiecriteria.

Unit tests zijn geschreven en slagen: Voor de nieuwe code zijn unit tests geschreven die alle belangrijke paden en functionaliteiten afdekken. Deze tests moeten zonder fouten slagen.

Code is gereviewed en goedgekeurd: De code is door een andere developer gereviewed en voldoet aan de standaarden voor leesbaarheid, onderhoudbaarheid en efficiëntie.

Geen openstaande bugs of issues: De functionaliteit moet zonder fouten werken volgens de specificaties, en er mogen geen openstaande kritieke bugs of problemen zijn.

Code is geïntegreerd in de main branch: De code moet correct geïntegreerd zijn in de hoofdcodbase, zonder merge-conflicten.

Documentatie is bijgewerkt: Indien nodig, is de documentatie van de code of de technische documentatie bijgewerkt met de wijzigingen.

### 7.2 Voor de gebruiker:

Een taak of functionaliteit wordt als "klaar" beschouwd door de gebruiker wanneer:

De functionaliteit werkt zoals verwacht zonder problemen: De gebruiker moet de functionaliteit kunnen gebruiken zoals beschreven in de user story, zonder onverwachte fouten of beperkingen.

De gebruikersinterface is intuïtief en gebruiksvriendelijk: Voor functies met een UI is de interface begrijpelijk en eenvoudig te gebruiken, zonder dat er een complexe uitleg of training nodig is.

De functionaliteit is grondig getest in een realistische omgeving: De functionaliteit is getest in een staging- of productieomgeving en werkt zonder problemen, zowel functioneel als qua prestatie.

Er is geen verdere feedback of verbeterpunten: De gebruiker heeft de functionaliteit getest en goedgekeurd, en er is geen aanvullende feedback of verzoek om wijzigingen meer nodig.

### 7.3 Algemeen voor het team:

Een taak of functionaliteit is pas volledig "klaar" wanneer:

Alle acceptatiecriteria van de user story zijn voldaan.

Functionaliteit is getest en gedocumenteerd: Alle testen (unit, integratie, acceptatie) zijn doorlopen en alle relevante documentatie is bijgewerkt.

De feature is gedemonstreerd aan de stakeholders: Het team heeft de functionaliteit gedemonstreerd aan de relevante stakeholders of product owner, en deze is goedgekeurd zonder verdere aanpassingen.

De functionaliteit is vrijgegeven voor gebruik: De functionaliteit is beschikbaar gesteld in de productieomgeving en klaar voor gebruik door eindgebruikers.



## 8. (Sprint) Planning

**Sprintduur:** Elke sprint duurt 2 weken.

**Team:** 2 developers werken aan de sprint.

### 8.1 Doel van Sprint Planning

Vaststellen welke user stories in de sprint worden opgenomen.

Focus op het leveren van waardevolle functionaliteiten.

### 8.2 Voorbereiding

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

### 8.3 Selectie van User Stories

Kies user stories die haalbaar zijn binnen de sprintduur.

Inschatting van de benodigde inspanning per story.

### 8.4 Taken Opstellen

User stories worden opgesplitst in kleinere, beheersbare taken.

Toewijzing van taken aan developers op basis van expertise.

### 8.5 Risico's en Obstakels

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

### 8.6 Definition of Done (DoD)

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

### 8.7 Commitment

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

## 9. Sprint Backlog

### 9.1 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.

### 9.2 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.

Schattingpunten: 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.

### 9.3 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.

### 9.4 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.