

Squad Health Check



Opdrachtgever

Team Rockstars IT

Groepscoach

Mestrom, Mark

Groepsbegeleider Opdrachtgever

Vente, Lesley

Groepsleden

Atakan Kocakoglu

Luc Lammers

Sjors Klaassen

Ties Vreeswijk

Zavid van den Worm

27-02-2024

Inhoud

Inleiding	3
Leeswijzer	3
Wie zijn wij?	3
Rockstars IT	4
Opdrachtbeschrijving.....	5
Analyse.....	6
User Stories	6
Requirements volgens MOSCOW	7
Use Cases	9
Context diagram	11
Conceptueel model.....	12
UI schetsen.....	13
Testplan	14
Architectuur	16
UML klassendiagram.....	17

Inleiding

Leeswijzer

In dit document wordt een analyse uitgevoerd op een software opdracht die wij hebben gekregen van Rockstars IT, onze opdrachtgever. Eerst wordt de groep voorgesteld en de opdrachtgever. Vervolgens is er een opdrachtbeschrijving om de opdracht duidelijk te maken. Vervolgens komen de verschillende onderdelen van de analyse. Om het af te sluiten eindigt het document worden conclusies vanuit de analyse getrokken waarop wij ons project gaan voortborduren.

Wie zijn wij?

Onze groep bestaat uit vijf leden, in dit hoofdstuk introduceert elk groepslid zichzelf.

Atakan Kocakoglu

Mijn naam is Atakan Kocakoglu, student HBO-ICT. Ik heb gekozen voor Fontys Hogeschool ICT, omdat dit me erg aansprak bij de open dagen van de school. Ik ben altijd ook geïnteresseerd geweest in het programmeren en in het ontwikkelen van apps en/of websites. Ik had al voorkennis over hoe ik websites kan maken, maar wist nog niets van al het andere in de ICT dus mijn doel was om dit te leren. Nu heb ik een goede keuze gemaakt om het software-profiel te kiezen en studeer ik hier met plezier.

Luc Lammers

Mijn naam is Luc Lammers, student aan de Fontys Hogescholen. Mijn interesse in ICT is ontstaan tijdens mijn tijd op het Philips van Horne in Weert, waar ik mijn Havo N&T diploma heb behaald. Ik ben gepassioneerd geraakt door programmeren en het opzetten van netwerken. Mijn programmeervaardigheden zijn voornamelijk gericht op Full-Stack development, met een voorkeur voor backend vanwege mijn plezier in het oplossen van problemen.

Sjors Klaassen

Mijn naam is Sjors Klaassen, ik ben 19 jaar oud en studeer ICT op Fontys. Ik heb VWO gedaan op Het College in Weert. Vervolgens ben ik naar de Technische Universiteit Eindhoven gegaan om Computer Science and Engineering te studeren, deze studie was uiteindelijk niet hoe ik hem verwacht had. Daarna heb ik de keuze gemaakt om naar HBO ICT te komen bij Fontys en hier studeer ik nu met veel plezier.

Ties Vreeswijk

Mijn naam is Ties vreeswijk. Ik ben 20 jaar oud en ik studeer ICT op Fontys Eindhoven. Ik kom van de Havo en ben daarna doorgegaan naar deze studie. Ik heb voor deze studie gekozen omdat ik het altijd al erg leuk heb gevonden om dingetjes te maken. Deze studie past daarom goed bij mij. Ik vind het wel lastig om mijn vinger te leggen op wat ik precies wil gaan kiezen, maar hier kom ik vanzelf wel achter.

Zavid van den Worm

Mijn naam is Zavid van den Worm. Ik ben een 17-jarige student aan Fontys, en ik volg de opleiding HBO-ICT. Ik heb vorig jaar mijn Havodiploma behaald, en ben toen meteen begonnen met de opleiding. Ik zit nu in het tweede semester van deze opleiding. Het is mijn doel om later een baan te krijgen in software-engineering, en/of front- en backend web development.

Rockstars IT

Team Rockstars IT is een bedrijf wat developers naar bedrijven stuurt om opdrachten uit te voeren. Rockstars regelt alles voor hun developers, zodat zij zich beter kunnen concentreren op de projecten. Bij Rockstars wordt er gewerkt met een 70/30 principe, waarbij 70% van de inkomsten die je genereert naar jou gaan en de overige 30% wordt gebruikt voor teambuilding en andere bedrijfszaken.

De opdracht die wij hebben gekregen van Rockstars is om een app te ontwikkelen waarmee de tevredenheid binnen squads (groepen van developers) wordt gemonitord.

Opdrachtbeschrijving

Wij als groep zijn uitgedaagd door Rockstars IT om een Squad Health Check app te maken. De Squad Health Check, SHC in het kort, is een manier om de werkervaring te meten van Rockstars bij verschillende werkplekken. Rockstars IT wil het formulier met vragen uit kunnen sturen naar de Rockstars. Vervolgens kunnen de Rockstars dit formulier invullen en terugsturen. Vervolgens kan de manager deze formulieren uitlezen en actie ondernemen als het nodig is. Deze web app bestaat uit twee onderdelen.

- Web app
- Backoffice

Elke groep maakt één van deze twee onderdelen. Onze groep maakt de web app. Op de web app moet de medewerker een vragenlijst krijgen van het bijbehorende bedrijf en deze in kunnen vullen. Per vraag kan de gebruiker doormiddel van een stoplicht een antwoord geven. De resultaten worden vervolgens teruggestuurd naar de backoffice.

Analyse

User Stories

Als gebruiker wil ik de Squad Health Check snel en in kunnen vullen op mijn telefoon omdat ik het dan gemakkelijk tussen werk door kan invullen.

Als gebruiker wil ik antwoord en een eventuele opmerking kunnen geven zodat de backoffice weet hoe mijn werkervaring is bij mijn werkplek en daarop kan hanteren.

Als gebruiker wil ik uitleg krijgen over de SHC omdat het kan zijn dat ik niet weet hoe het werkt en wat het nut van de SHC is.

Als gebruiker wil ik het invullen kunnen onderbreken om later weer verder te gaan, zodat mijn vooruitgang niet verloren gaat als ik aan iets moet werken.

Als gebruiker wil ik gemakkelijk tussen vragen kunnen wisselen als ik terug wil naar vorige vragen om mijn antwoorden aan te passen.

Als gebruiker wil ik na het invullen van de SHC een overzicht zien van mijn antwoorden en deze eventueel ook nog kan aanpassen voor ik alles inlever. Op deze manier weet ik zeker dat alles goed is ingevuld.

Als gebruiker wil ik een goed ontworpen user interface, zodat het invullen van de SHC gemakkelijker en fijner verloopt.

Als gebruiker wil ik extra uitleg bij de vragen krijgen, zodat er geen verwarring ontstaat over de vraag.

Als Rockstars IT wil ik de antwoorden van de vragenlijst teruggestuurd krijgen in de database zodat ik deze kan uitlezen.

Requirements volgens MOSCOW

Must-haves

FR1. De app moet bij de uitnodigingslink horende vragenlijst ophalen uit de database.

B-1.01 Er kan maar één soort vragenlijst ingevuld worden via de link.

B-1.02 De link is niet meer te gebruiken na 1x.

K-1.01 De vragenlijst die bij de gebruiker hoort opent automatisch.

FR2. De gebruiker moet bij elke vraag een antwoord kunnen geven.

B-2.01 De gebruiker kan kiezen van drie opties: rood, oranje en groen.

B-2.02 Zodra de gebruiker een antwoord heeft gekozen kan deze niet meer leeg worden gemaakt.

FR3. Ingevulde vragenlijst moet terug worden gestuurd naar de database.

B-3.01 Zodra de vragenlijst verstuurd is naar de database kan deze niet meer worden aangepast.

K-3.02 De vragenlijst wordt na iedere aanpassing geüpdatet in de database.

NFR4. App moet responsive zijn op een telefoon-formaat.

Should-haves

FR5 Er is een uitleg van elk punt waarop de gebruiker antwoord geeft.

FR6 De gebruiker kan achteraf nog een antwoord van een vraag veranderen.

K-6.01 Als de gebruiker de vraag wil veranderen staat de vraag op de oude antwoorden.

FR7 De gebruiker kan een overzicht zien van alle vragen.

K-7.01 De gebruiker kan via dit overzicht naar de vragen navigeren.

K-7.02 De gebruiker kan de gegeven antwoorden zien bij elke vraag.

FR8 De app geeft een melding als de vragenlijst niet compleet is ingevuld.

K-8.01 Als de vragenlijst niet is ingevuld geeft de app een melding welke vragen niet ingevuld zijn.

NFR9 De app wordt publiek gehost.

Could-haves

NFR10 De app is Engelstalig.

FR11 Gebruiker kan een opmerking toevoegen bij het antwoord.

NFR12 App kan vertaald worden naar verschillende talen.

NFR13 App moet responsief zijn op alle platformen.

FR14 Vragen kunnen "flaggen" om later op terug te kunnen komen

K-14.01 Bij het overzicht kan de gebruiker zien welke vragen "geflagd" zijn.

FR15 als ingelogde gebruiker kun je naar de webapp gaan om te zien welke questionnaires je nog moet invullen

Use Cases

Naam	UC1: Vragen ophalen vanuit link
<i>Samenvatting</i>	Wanneer een actor op de invitelink klikt, wordt deze naar de webapp genavigeerd.
<i>Actors</i>	Squad member
<i>Aannames</i>	De actor heeft een uitnodigingslink ontvangen via de mail.
<i>Scenario</i>	<ol style="list-style-type: none"> 1. De actor klikt op de ontvangen uitnodigingslink. [1] 2. Het systeem navigeert naar de web app. 3. De landingspagina geeft een beschrijving en uitleg over de vragenlijst.
<i>Uitzonderingen</i>	<ol style="list-style-type: none"> 1. De uitnodigingslink werkt niet. De actor stuurt een mail naar de verstuurder om aan te geven dat de link niet werkt.
<i>Resultaat</i>	De actor komt uit op de web app

Naam	UC2: Antwoord geven
<i>Samenvatting</i>	Als een actor een vraag heeft gelezen geeft deze een antwoord en dit antwoord wordt in de database opgeslagen.
<i>Actors</i>	Squad member
<i>Aannames</i>	De actor snapt wat de vraag inhoudt. De actor heeft een unieke link
<i>Scenario</i>	<ol style="list-style-type: none"> 1. Het systeem toont een pagina waarop de actor een vraag kan lezen en beantwoorden. 2. De actor kiest een van de drie kleuren om antwoord te geven op deze vraag. (Zie B-M2.01) [1] 3. De actor klikt op volgende vraag en het systeem slaat het antwoord op in de database. 4. Het systeem toont de volgende vraag.
<i>Uitzonderingen</i>	<ol style="list-style-type: none"> 1. De actor heeft de vraag niet ingevuld. Het systeem slaat de vraag niet op. UC gaat verder.
<i>Resultaat</i>	De vraag is opgeslagen.

Naam	UC3: Antwoord veranderen
<i>Samenvatting</i>	Als een actor een eerder ingevulde vraag wil aanpassen, kan deze teruggaan naar de vraag en het antwoord vervolgens aanpassen.
<i>Actors</i>	Squad member
<i>Aannames</i>	De actor is al ooit bij deze vraag geweest. De actor heeft een unieke link
<i>Scenario</i>	<ol style="list-style-type: none"> 1. De actor navigeert via het overzicht of de knoppen bij de vragen naar de vraag waarvan de actor het antwoord wilt veranderen. 2. Het systeem navigeert naar de pagina. 3. Het systeem gaat naar UC2.[1]
<i>Uitzonderingen</i>	<ol style="list-style-type: none"> 1. Het antwoord wordt niet veranderd. UC2 wordt daarom niet uitgevoerd.
<i>Resultaat</i>	Het nieuwe antwoord wordt opgeslagen.

Naam	UC4: Naar vragen overzicht navigeren
<i>Samenvatting</i>	De actor kan vanuit de vragen navigeren naar het vragen overzicht door middel van een knop. Hier kan de gebruiker een lijst van de vragen zien en er snel naartoe navigeren. De actor kan ook zijn ingevulde antwoorden zien.
<i>Actors</i>	Squad member
<i>Aannames</i>	De actor heeft toegang tot de webapp. De actor weet welke knop naar het overzicht navigeert.
<i>Scenario</i>	<ol style="list-style-type: none"> 1. De actor wil naar het vragenoverzicht navigeren en klikt op de bijbehorende knop. 2. Het systeem navigeert naar de overzicht pagina.
<i>Uitzonderingen</i>	
<i>Resultaat</i>	De actor is op de overzicht pagina.

Naam	UC5: Vragenlijst controleren en versturen
<i>Samenvatting</i>	Als de actor alle vragen heeft ingevuld laat het systeem een overzicht zien van alle vragen waarin de gebruiker terug kan gaan naar vragen.
<i>Actors</i>	Squad member
<i>Aannames</i>	De actor heeft toegang tot de web app. De actor is klaar met de vragen in te vullen.
<i>Scenario</i>	<ol style="list-style-type: none"> 1. Het systeem laat een pagina zien met een overzicht van de vragen. Bij deze vragen kan de actor de gegeven antwoorden zien. Op de pagina staat ook een knop om de vragenlijst te versturen. 2. De actor controleert al de gegeven antwoorden. Als de actor tevreden is met de gegeven antwoorden klikt de actor om de antwoorden te versturen. [1] 3. Het systeem zet de "isUsed" in de link op true in de database. 4. Het systeem geeft een melding waarin de actor ziet dat het gelukt is.
<i>Uitzonderingen</i>	<ol style="list-style-type: none"> 1. Niet alle vragen zijn ingevuld. Het systeem verandert de knop om de vragenlijst te versturen zodat deze niet meer gebruikt kan worden. Het systeem geeft ook een tekst boven de knop waarin staat welke vraag niet ingevuld is. De actor klikt op de niet ingevulde vraag en voert UC1 uit. Alle vragen zijn ingevuld en ga verder met use case.
<i>Resultaat</i>	De ingevulde vragen zijn verstuurd naar de database.

Sprintdoel:

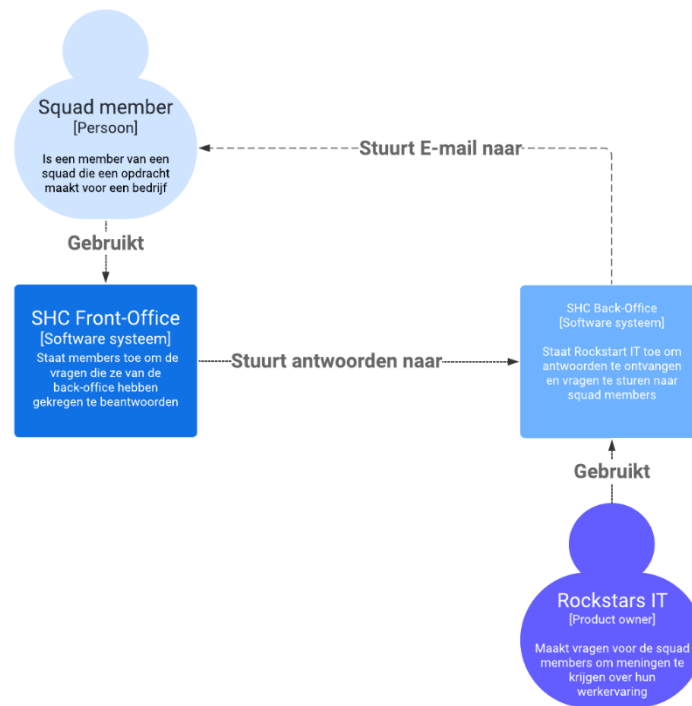
Requirements bijwerken en in trello zetten

Nadenken wat voor elke requirement nodig is (bijv tabel front-end etc), hiermee kunnen we beter tijd inschatten

Context diagram

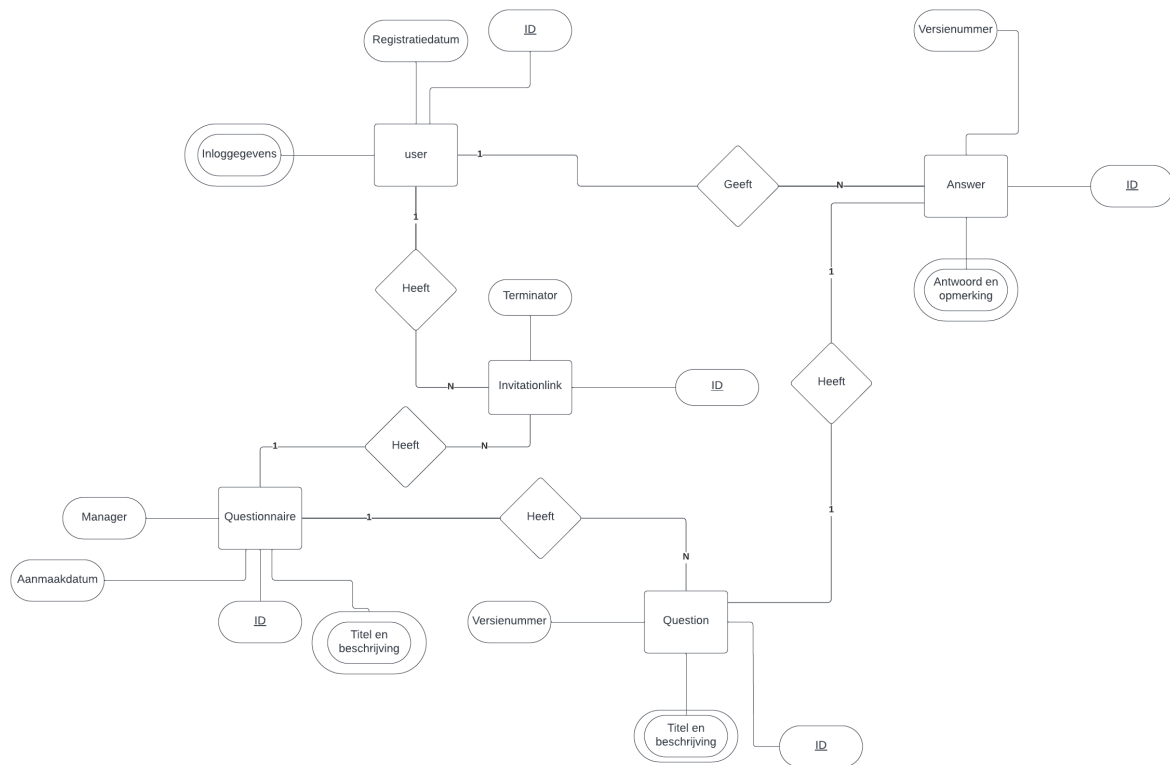
In de onderstaande afbeelding kun je een context diagram zien van het gehele project. Hier kun je zien hoe alle delen van het gehele project met elkaar samen werkt. Het begint bij Rockstars IT, zij maken vragen zodat ze meningen krijgen van de squad members over hun werkervaring. Ze gebruiken de Back-office om vragen te maken en door te sturen naar de squad members. De squad members beantwoorden de met de front-office en kunnen dit ook zo terug sturen.

Context Diagram SHC



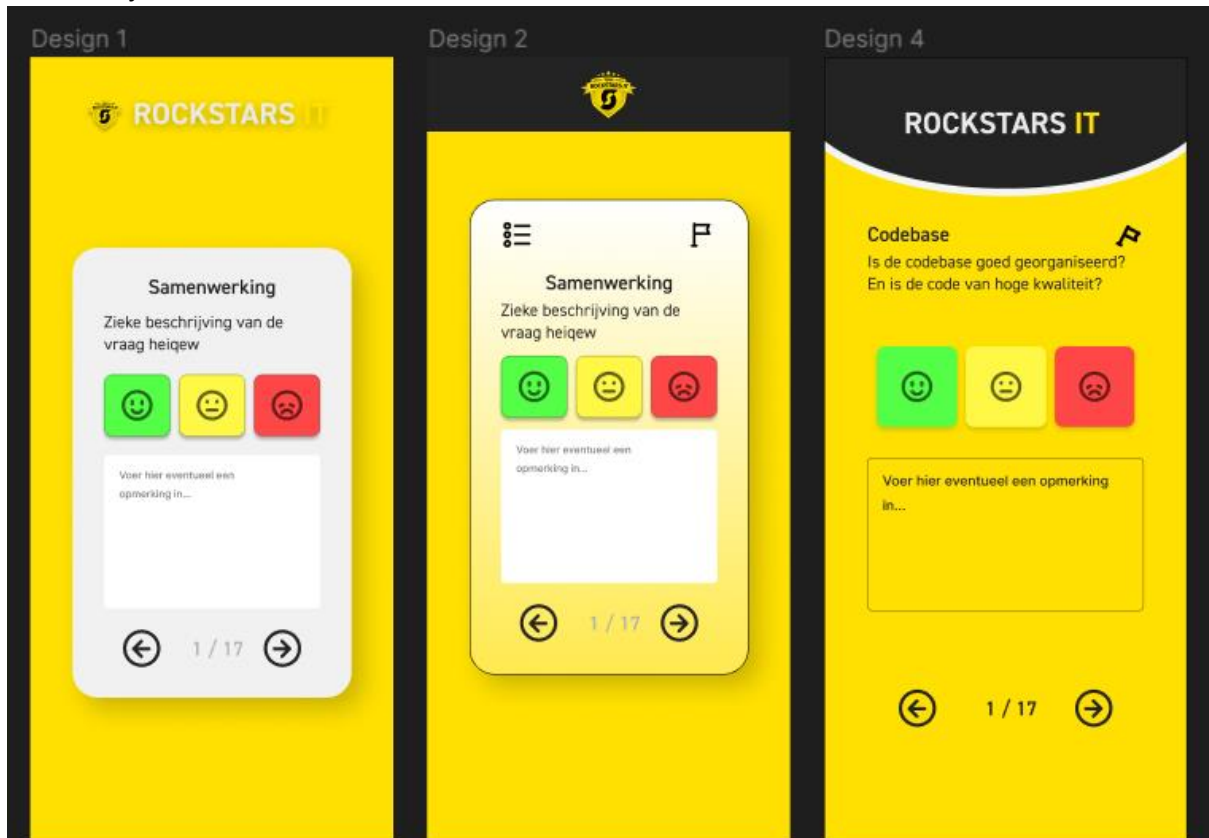
Conceptueel model

In de onderstaande afbeelding is het conceptueel model te zien van de front-office. Hier kun je de entiteiten zien die worden gebruikt met de bijbehorende attributen. Dit heeft een overzichtelijk beeld over hoe de verschillende entiteiten met elkaar verbinden.

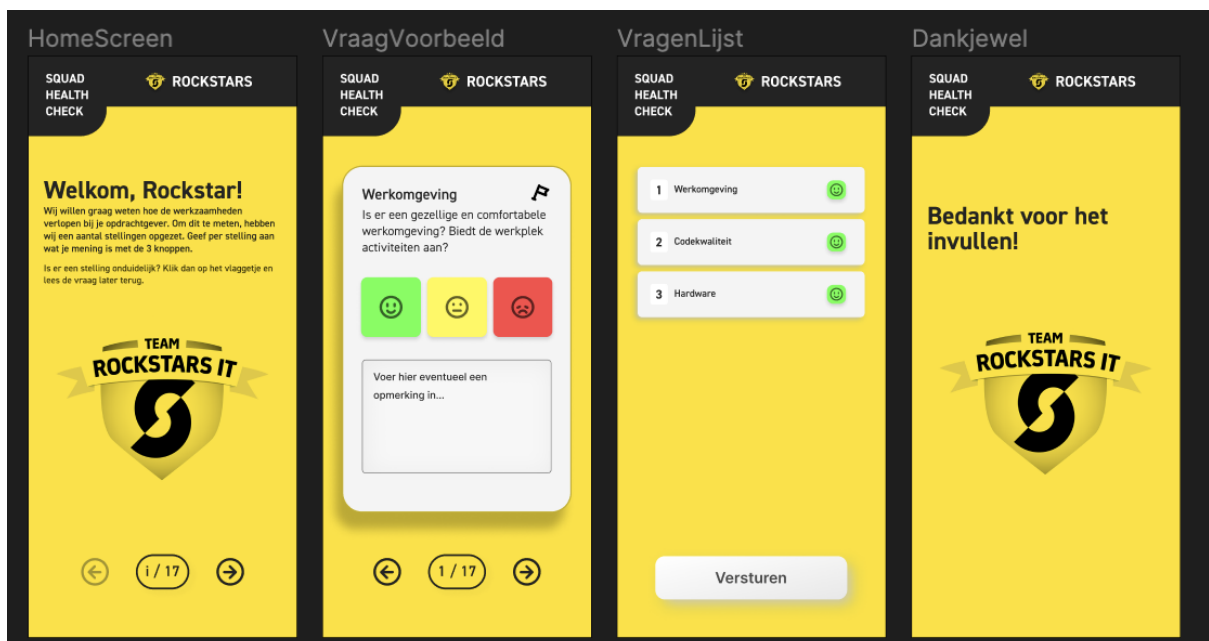


UI schetsen

Voor het project hebben wij een aantal UI-schetsen moeten maken. Deze zijn door meerdere iteraties heen gegaan, en wij hebben voor elke iteratie aan verschillende mensen voor feedback gevraagd. Deze informatie hebben wij gebruikt om onze ontwerpen te verbeteren. De vorige iteraties zijn hieronder te zien.



Na het verwerken van de gekregen feedback hebben wij een aantal aanpassingen gemaakt, en zijn uiteindelijk op de volgende ontwerpen uitgekomen.



Testplan

Test case	Use Case(s)	Prioriteit (1 t/m 5)	Invoer	Verwachte uitvoer
TC01	UC1	5	Link naar pagina	Questionnaire pagina opent.
TC02	UC2	2	Kleurkeuze: "rood" Omschrijving: "ik vind het niet leuk" Vraag Id: "1"	Op de overzicht pagina staat bij vraag 1 de keuze rood en de geschreven omschrijving.
TC03	UC2	4	Kleurkeuze: "" Omschrijving: "Ik vind het heel fijn werken" Vraag Id: "1"	De beschrijving wordt opgeslagen en de gebruiker kan door naar de volgende vraag.
TC04	UC3	3	Oude kleurkeuze: "rood" Nieuwe Kleurkeuze "geel" Oude omschrijving: "het werkte niet fijn" Nieuwe Omschrijving: "het valt toch wel mee" Vraag Id: "1"	Het antwoord bij vraag 1 is veranderd naar geel, en de omschrijving is veranderd naar: "het valt toch wel mee".
TC05	UC3	3	Oude kleurkeuze: "groen" Nieuwe kleurkeuze: "groen" Oude omschrijving: "Het werkte heel goed" Nieuwe omschrijving: "Het werkte heel goed" Vraag Id: "3"	De antwoorden bij vraag 3 blijven hetzelfde als ze voorheen waren.
TC06	UC4	4	ButtonClick: "GoToSummary"	Overzicht pagina opent met ingevulde vragen.
TC07	UC5	4	Omschrijving: "Vragenlijst inleveren" Vragenlijst: "compleet ingevuld"	De vragenlijst staat opgeslagen in de database en de gebruiker krijgt te weten dat het gelukt is.
TC08	UC5	3	ButtonClick: "StuurVragenlijst" Vraag Id "3, 6, 9" niet ingevuld	De gebruiker krijgt een melding waarin staat dat vraag 3, 6, 9 niet ingevuld zijn.

Testmatrix

	FR1	FR2	FR3	FR6	FR7	FR8
TC01	X					
TC02		X				
TC03		X				
TC04				X		
TC05				X		
TC06					X	
TC07			X			X
TC08			X			X

Architectuur

Het project wordt gebouwd op basis van de N-tier architectuur. De N-tier architectuur, beter bekend als de meerlaagse architectuur, is een architectuur die gebaseerd is op lagen. Elk van deze lagen heeft een eigen verantwoordelijkheid. Deze lagen zijn onderling verbonden, maar kunnen niet overgeslagen worden. De bovenste laag communiceert dus niet direct met de onderste laag. De voordelen van het gebruiken van de N-tier architectuur zijn als volgt:

- **Schaalbaarheid:** Er kunnen altijd meer lagen worden toegevoegd. Op deze manier kan het programma altijd uitgebreider worden.
- **Modulariteit:** De N-tier architectuur verdeelt de applicatie in meerdere lagen. Dit maakt het modulair en gemakkelijk te onderhouden.
- **Herbruikbaarheid:** Omdat de applicatie is geschreven in meerdere lagen kunnen deze ook herbruikt worden in andere situaties.
- **Onderhoudbaarheid:** De modulariteit en scheiding van verantwoordelijkheden maakt het makkelijker om aanpassingen te maken.
- **Platformonafhankelijkheid:** Omdat de lagen allemaal een eigen verantwoordelijkheid hebben kan het platform waarop de applicatie uiteindelijk werkt verschillend zijn. Het zorgt ervoor dat dezelfde applicatie op bijvoorbeeld web, dekstop of telefoonapps te gebruiken zijn.

In de N-tier architectuur zijn er verschillende soorten lagen. De volgende lagen komen terug in de SHC applicatie.

- **Presentatie laag:** Dit is de bovenste laag. Deze laag is verantwoordelijk voor het presenteren van informatie aan de gebruiker en het verwerken van input. In ons geval wordt dit een ASP.NET core Blazor project.
- **Applicatie laag:** Deze laag bevat de business logica van de applicatie. Deze laag verwerkt de verzoeken van de presentatie laag, en voert daar vervolgens logica op uit. Deze laag moet logica uit kunnen voeren zonder details van de presentatie laag en data laag te weten.
- **Data laag:** Dit is de onderste laag van de applicatie. Deze laag is verantwoordelijk voor het ophalen, plaatsen en aanpassen van data in de database. De SHC-applicatie maakt gebruik van een relationele database, een SQL database.

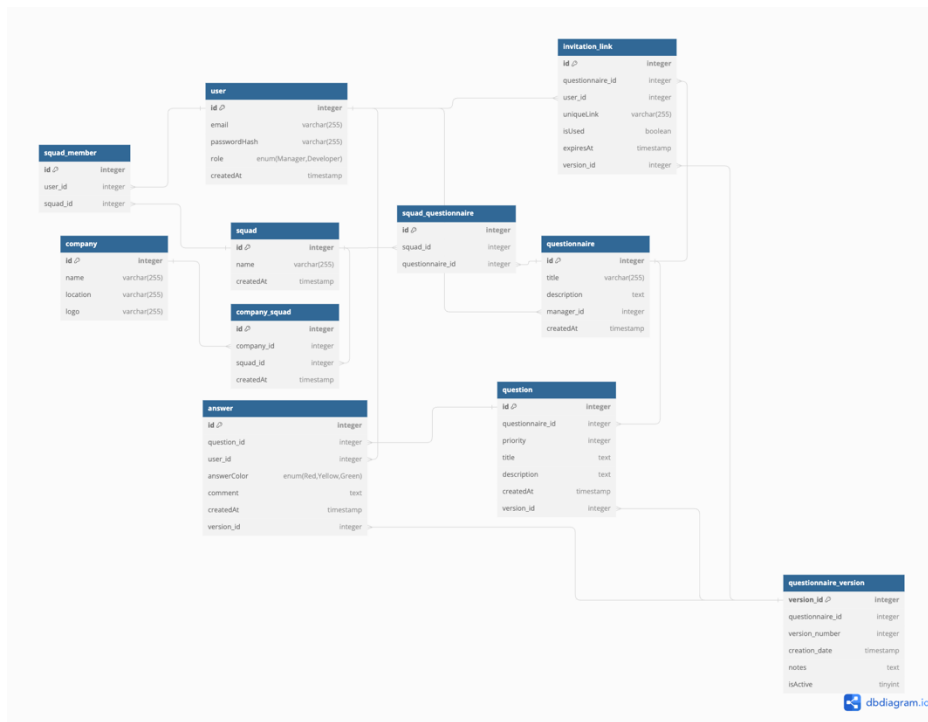
Omdat dit project alleen interactie heeft met een gebruiker en een database zijn er niet meer lagen dan dit nodig. De gebruiker heeft interactie met de presentatie laag. De applicatie laag voert vervolgens logica uit op deze interactie. De verwerkte input wordt via de data laag in de database verwerkt. Er zijn geen andere systemen waar de applicatie gebruik van maakt, en daarom is het beter om een simpele architectuur aan te houden. Omdat N-tier schaalbaar is kan er achteraf bij uitbreidingen altijd nog een nieuwe laag worden toegevoegd.

ERD

Samen met de andere groepen is er een database ontworpen die er als volgt uitziet. Deze is ontworpen voor het hele project en gemaakt voor mogelijke uitbreiding in de toekomst.

Het gedeelte van de frontoffice is alles rechts van “squad_questionnaire” inclusief “answer”. Dit zijn de tabellen die wij moeten kunnen bereiken in de database en wij ook gebruik van zullen maken.

‘Alles links van “squad_questionnaire” is gemaakt voor de backoffice zodat er beheer van de squads en questionnaires kan plaatsvinden.



UML-klassendiagram

