

Startt learningstories

Ik wil aantonen dat ik betrouwbare bronnen gebruik.

Situatie: Ik moet leren hoe ik betrouwbare bronnen gebruik zodat ik het kan implementeren in mijn werk.

Taak: De taak was om betrouwbare bronnen te vinden. Ook de bronnen correct te vermelden in een bronvermelding.

Aanpak: Ik heb in de knowledgebase bekeken hoe je correct een bron kan vermelden en hoe je een betrouwbare bron opzoekt.

Resultaat: Ik kan nu een bron vinden en beoordelen of het betrouwbaar is. Ook kan ik het correct vermelden in een bronvermelding.

Reflectie: Ik had al een idee hoe ik een betrouwbare bron kan vinden, maar toen ik in het knowledgebase had gekeken had ik een sterker idee. Ook heb ik eerdere ervaringen gehad hoe ik een bron vermeld in een bronvermelding. In de knowlegdebase staan er handige tips hoe ik sneller een bron kan vinden.

Transfer: Ik wil alles dat ik heb geleerd heb over bronnen kunnen meenemen naar het volgende blok.

Startt met learning stories

Ik wil leren hoe ik een rationale database ontwerp en implementeer.

Situatie: Ik wil leren hoe ik een rationale database ontwerp en implementeer.

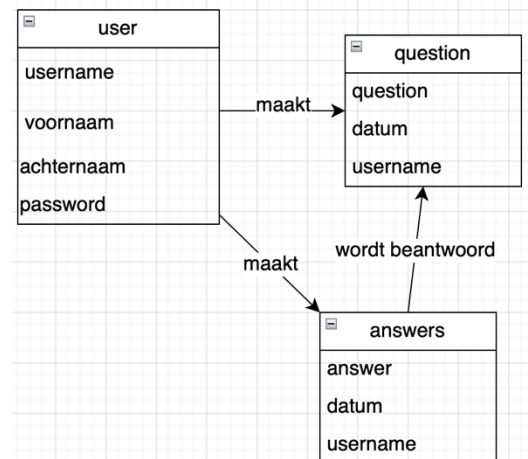
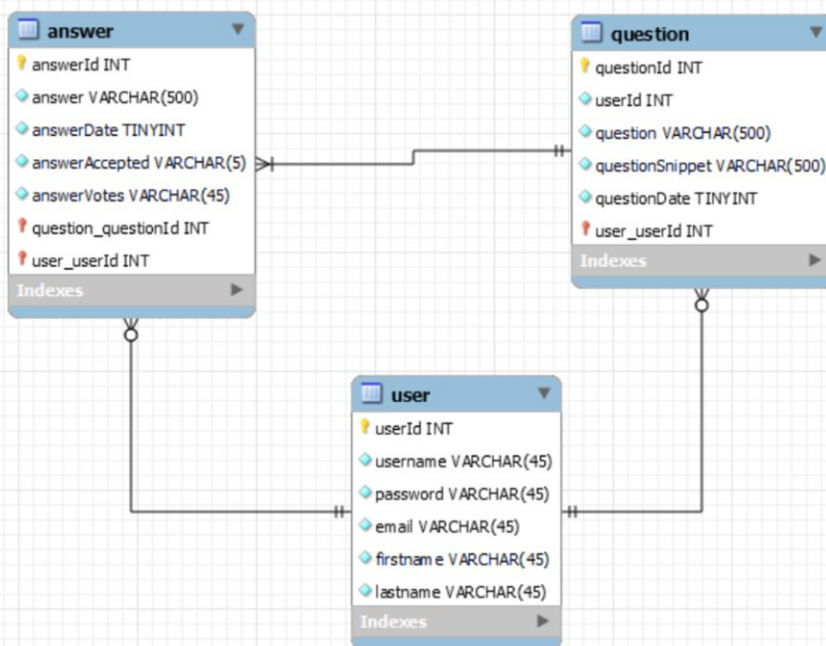
Taak: De taak was om een rationale database te ontwerpen en implementeren in de database.

Aanpak: Ik ging eerst naar de knowledgebase om info te krijgen over hoe je een database überhaupt ontwerpt. In de knowledgebase stond dat je eerst een schets moet maken en dan pas een ERD. Hier onder heb ik een foto met de schets en ERD. Ik heb dan een class diagram gemaakt. En daarna de database ontwerp.

Resultaat: Ik heb een nu een rationale database ontworpen. De ERD zit ook in de database.

Reflectie: Ik vond de class diagram maken het makkelijkste, maar toen ik de ERD moest maken ging dat moeilijker. De lijnen trekken tussen de columns waren beetje moeilijk dus ik moest via een youtube video leren hoe je de foreign keys moest rationaliseren.

Transfer: Ik wil wat ik geleerd heb over ERD meenemen naar de volgende blok.



Start met learningstories

Ik wil aantonen dat ik constructief kan samenwerken.

Situatie: Ik moet kunnen aantonen dat ik constructief kan samenwerken.

Taak: De taak was om een samenwerkingscontract te maken met mijn partner. Ook dat ik constructief feedback moet geven en krijgen.

Aanpak: Wij gingen als eerst het samenwerkingscontract ondertekenen. Daarna moesten wij constructieve feedback geven. In het knowledgebase stond hoe je iemand een goede feedback kon geven en krijgen.

Resultaat: Wij hebben een samenwerkingscontract. Ook heb ik feedback gegeven. Ik heb ook feedback gekregen over mijn code.

Reflectie: Het contract maken ging goed. Ik heb feedback gegeven en ook ontvangen. Die heb ik dan toegepast in mijn code. Ook waren wij heel open voor feedback voor een ander.

Transfer: Ik wil graag de feedback die ik heb gekregen meenemen naar het volgende blok. Ook wil ik het geven van constructief feedback goed beheersen en toepassen in het volgende blok.

Startt met learningstories

Ik wil leren hoe ik mijn project beschrijf met de unified modeling language

Situatie: Ik moet mijn project beschrijven met UML.

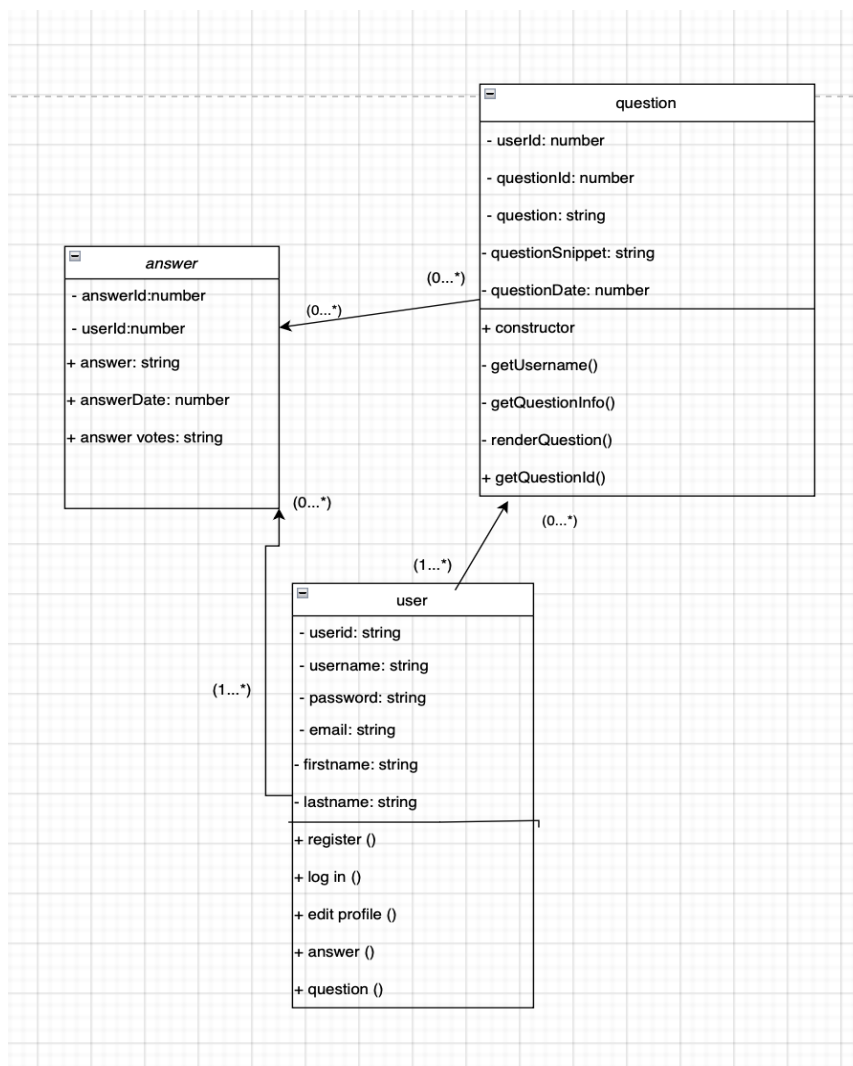
Taak: De taak was om een UML te maken. Met de UML moet je je code beschrijven.

Aanpak: Om een UML te maken moet je over class diagrams leren. Dus ik ging naar de knowledgebase om er meer erover te weten. Toen ik een idee had hoe ik een class diagram kon maken, ging ik daarna de UML maken. Ook had ik naar een paar voorbeelden gekeken om een beter idee te krijgen

Resultaat: Ik heb een nu een UML die mijn code beschrijft. De foto is eronder.

Reflectie: De UML maken ging moeilijk, want je moet je code goed begrijpen voordat je het in een UML moet zetten. Je mag ook geen onnodige code plaatsen in de UML. Dat vond ik het moeilijkste eraan. Verder was het maken van de UML goed gegaan.

Transfer: Ik wil wat ik geleerd heb van UML en class diagram meenemen naar het volgende blok, zodat ik een duidelijk overzicht heb van mijn code.



Startt met learningstories

Ik wil leren hoe ik de basisconcepten van OOP toepas

Situatie: Ik moet leren hoe ik de basisconcepten van OOP gebruik in mijn code.

Taak: De taak was OOP te kunnen toepassen in mijn code en leren wat OOP is.

Aanpak: Mijn aanpak was om eerst de knowledgebase te bekijken. Het was wel handig, maar ik had meer info nodig. De online les van paar weken geleden had wat uitleg van OOP. Een docent had OOP gedemonstreerd. Ik had ernaar gekeken en probeerde het in mijn code.

Resultaat: Ik heb nu OOP in mijn code en kan OOP uitleggen. Ik kan een class beschrijven en ik begrijp wat een scope inhoudt. Ik heb een foto van mijn code hieronder van de constructor en classes. Ik heb gebruik gemaakt van classes en constructors. Ik heb ook geprobeerd om in mijn code de verantwoordelijkheden te scheiden.

Reflectie: Ik vond het in het begin moeilijk om met OOP te kunnen werken, omdat ik niet wist hoe ik OOP moest toepassen aan mijn code. Ik vond mijn code ingewikkeld. Dan moest ik mijn code ook nog in OOP toepassen. Dat ging moeizaam. Het begrijpen van OOP was makkelijk. Het toepassen daarvan was moeilijk.

Transfer: Ik wil wat ik geleerd heb van OOP later kunnen toepassen en versterken in het volgende blok.

Hier onder is mijn class question en mijn constructor.

```
class Question {  
    private userId: number;  
    private questionId: number;  
    private question: string;  
    private questionSnippet: string;  
    private questionDate: number;  
  
    public constructor(userId: number, questionId: number,  
        this.userId = userId;  
        this.questionId = questionId;  
        this.question = question;  
        this.questionSnippet = questionSnippet;  
        this.questionDate = questionDate;  
}
```

Startt met learningstories

Ik wil leren hoe ik het OO-principe encapsulation toepas op mijn project.

Situatie: Ik moet leren hoe ik encapsulation in mijn code moet gebruiken. Ik moet ook leren wat encapsulation is.

Taak: Ik moet in mijn code access modifiers zetten zoals, public, private en protected. Ik moet ook gebruik maken van getters and setters.

Aanpak: Ik had naar de knowledgebase gekeken, daar stond wat encapsulation inhoudt. Ook keek ik naar het internet over wat een private, public en protected class is. Daarna keek ik naar mijn code en had ik een private class gemaakt. Ook bij de constructor heb ik een public class gemaakt.

Resultaat: In mijn code zit encapsulation. Er zijn ook getters and setters in mijn code. Ik heb een foto van de public en private class.

Reflectie: Het encapsulation van mijn code was moeilijk, want er waren geen voorbeelden in de knowledgebase. Ik moest het gaan opzoeken. Getters and setters moest ik ook gaan opzoeken.

Transfer: Ik wil de encapsulation principe van OOP meenemen naar het volgende blok, zodat ik beter code overzicht heb.

Hier zie je de private question en public constructor.

```
class Question {  
    private userId: number;  
    private questionId: number;  
    private question: string;  
    private questionSnippet: string;  
    private questionDate: number;  
  
    public constructor(userId: number, questionId: number,  
        this.userId = userId;  
        this.questionId = questionId;  
        this.question = question;  
        this.questionSnippet = questionSnippet;  
        this.questionDate = questionDate;  
}
```

Startt met learningstories

Ik wil leren hoe ik het OO-principe abstraction toepas op mijn project.

Situatie: Ik moet leren hoe ik abstraction toepas in mijn code. Ook moet ik leren wat abstraction is.

Taak: Ik moet een klasse ontwerpen met attributen en methodes. Dat moet ik dan toepassen in mijn code. Ik moet leren wat abstraction is.

Aanpak: Ik ging naar de knowledgebase om meer te weten te komen over abstraction. Na dat ik heb gelezen, ging ik opzoeken op het internet voor meer info. Daarna ging ik mijn code veranderen.

Resultaat: Ik heb abstraction in mijn code. Ik heb attributen en methodes in mijn code.

Reflectie: Ik vond abstraction ook moeilijk toe te passen, want er zijn geen voorbeelden. Je moet ook zelf opzoeken wat het best bij abstraction past. Ook het methodes maken en attributen maken is moeilijk, want je moet zelf bepalen wat belangrijk is en wat niet. Daarna moet je het ook toepassen aan je code. Uiteindelijk is het wel gelukt om het toe te passen.

Transfer: Ik wil het abstraction principe meenemen naar het volgende blok, zodat ik een overzichtelijk code heb.

Hier is de code met een class, methode en attributen. Je kan ook de abstraction zien.

```
class Question {
  private userId: number;
  private questionId: number;
  private question: string;
  private questionSnippet: string;
  private questionDate: number;

  public constructor(userId: number, questionId: number, question: string, questionSnippet: string, questionDate: number) {
    this.userId = userId;
    this.questionId = questionId;
    this.question = question;
    this.questionSnippet = questionSnippet;
    this.questionDate = questionDate;
  }

  private async getUsername(): Promise<string | undefined> {
    try {
      const userData: any = await api.queryDatabase("SELECT username FROM user WHERE userId = ?", [this.userId]);
      return userData && userData.length > 0 ? userData[0].username : undefined;
    } catch (error) {
      console.error("Error fetching user data:", error);
      return undefined;
    }
  }
}
```

Startt met learningstories

Ik wil leren hoe ik prototyping gebruik om de gebruikersinterface van mijn product vorm te geven

Situatie: Ik moet leren hoe ik prototypes maak voor een gebruikersinterface.

Taak: ik moet een task flow uitwerken in een schets met de user story nummer 13#. Ook moet ik een lofi schets maken van de user story.

Aanpak: Ik moest opzoeken in de database wat een prototype is en hoe het eruitziet. Ook moest ik opzoeken wat een taskflow was. Daarna begon ik met schetsen op papier en digitaal.

Resultaat: Ik heb nu een paar schetsen van de userstory. Ik heb de taskflow op papier. Ik heb een medium fidelity van een wireflow digitaal gemaakt. Hier onder foto's.

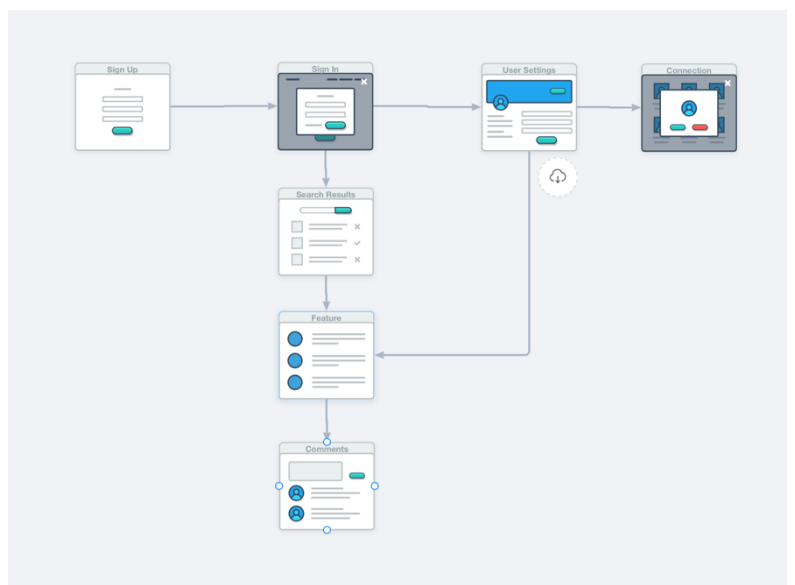
Reflectie: Ik vond het ontwerpen van prototypes leuk, want je kan je ontwerp zien tot leven komen. Ook heb je een duidelijk overzicht wat er allemaal moet gebeuren in je code.

Transfer: Ik wil het ontwerpen van prototypes meenemen naar het volgende blok.

Start → Login → Home page → Profile →

My questions → click on question →

question with answers



Ik wil leren hoe ik de behoeftes van de gebruiker vastleg in een gebruikersonderzoek.

Leerkaart	
Student of team naam	Studentnummer(s)
Naam Salina Muhammad	123 500921293
Inzicht naam	Inzicht datum
Test naam Question maken	week 5
1. HYPOTHESE	
Ik dacht dat... (zie testkaart)	
<u>het</u> maken van een question gemakkelijk gaat, omdat de website aan het werken is.	
2. OBSERVATIE	
Bijvoorbeeld "Niet iedereen kon de menuknop vinden, 2 van de 3 mensen klikten eerst op een andere knop"	
Resultaat tabellen en foto's kunnen aan de rechterkant onder 'documentatie' worden geplaatst.	
Het kostte veel clicks voordat iemand een question kon maken. Er was geen link om <u>naar</u> de question form te gaan. Je moet via de profile een question maken.	
3. LEERPUNT EN INZICHT	
Bijvoorbeeld: Het symbool voor de menuknop is te onduidelijk	
Link moet erbij komen in de <u>homepage</u> , zodat je een vraag kan maken.	
4. ADVIES	
Bijvoorbeeld: Andere iconen voor menu zoeken en deze aan mensen voorleggen om te zien wat het beste werkt	

Leerkaart	
Student of team naam	Studentnummer(s)
Naam Salina Muhammad.	123 500921293
Inzicht naam	Inzicht datum
Test naam Question maken	week 5
1. HYPOTHESE	
Ik dacht dat... (zie testkaart)	
De question maken soepel gaat, want de website is duidelijk. Dus je kan gemakkelijk <u>navigeren</u>	
2. OBSERVATIE	
Bijvoorbeeld "Niet iedereen kon de menuknop vinden, 2 van de 3 mensen klikten eerst op een andere knop"	
Resultaat tabellen en foto's kunnen aan de rechterkant onder 'documentatie' worden geplaatst.	
Je kan wel registreren. Er waren weinig fouten gemaakt. Ook questions maken ging goed.	
3. LEERPUNT EN INZICHT	
Bijvoorbeeld: Het symbool voor de menuknop is te onduidelijk	
Ik moet een link maken voor de vraag page. Verder is het goed.	
4. ADVIES	
Bijvoorbeeld: Andere iconen voor menu zoeken en deze aan mensen voorleggen om te zien wat het beste werkt	

Testkaart

Student of team naam

Studentnummer(s)

Naam Salina Muhammad

500921293

Test naam

Test deadline

Effectiviteit Efficiënt question maken

week 5

1. HYPOTHESE

Ik denk dat...

het maken van een question gemakkelijk gaat, omdat de website aan het werken is.

2. TEST

Om dit te testen...

Moet de gebruiker registreren en een question maken.

|

3. METRICS

Zie metrics

Ik meet hoe veel clicks het duurt en hoelang het duurt.

4. CRITERIA

De hypothese moet gevalideerd worden als...

De hypothese moet verworpen worden als...

de question in 60 seconden gemaakt word.

de question meer dan 60 seconden gemaakt word.

Testkaart

Student of team naam

Studentnummer(s)

Naam Salina Muhammad

123 500921293

Test naam

Test deadline

Effectiviteit Question maken

week 5

1. HYPOTHESE

Ik denk dat...

De question maken soepel gaat, want de website is duidelijk. Dus je kan gemakkelijk navigeren.

2. TEST

Om dit te testen...

Moet de persoon registreren en een question maken.

3. METRICS

Zie metrics

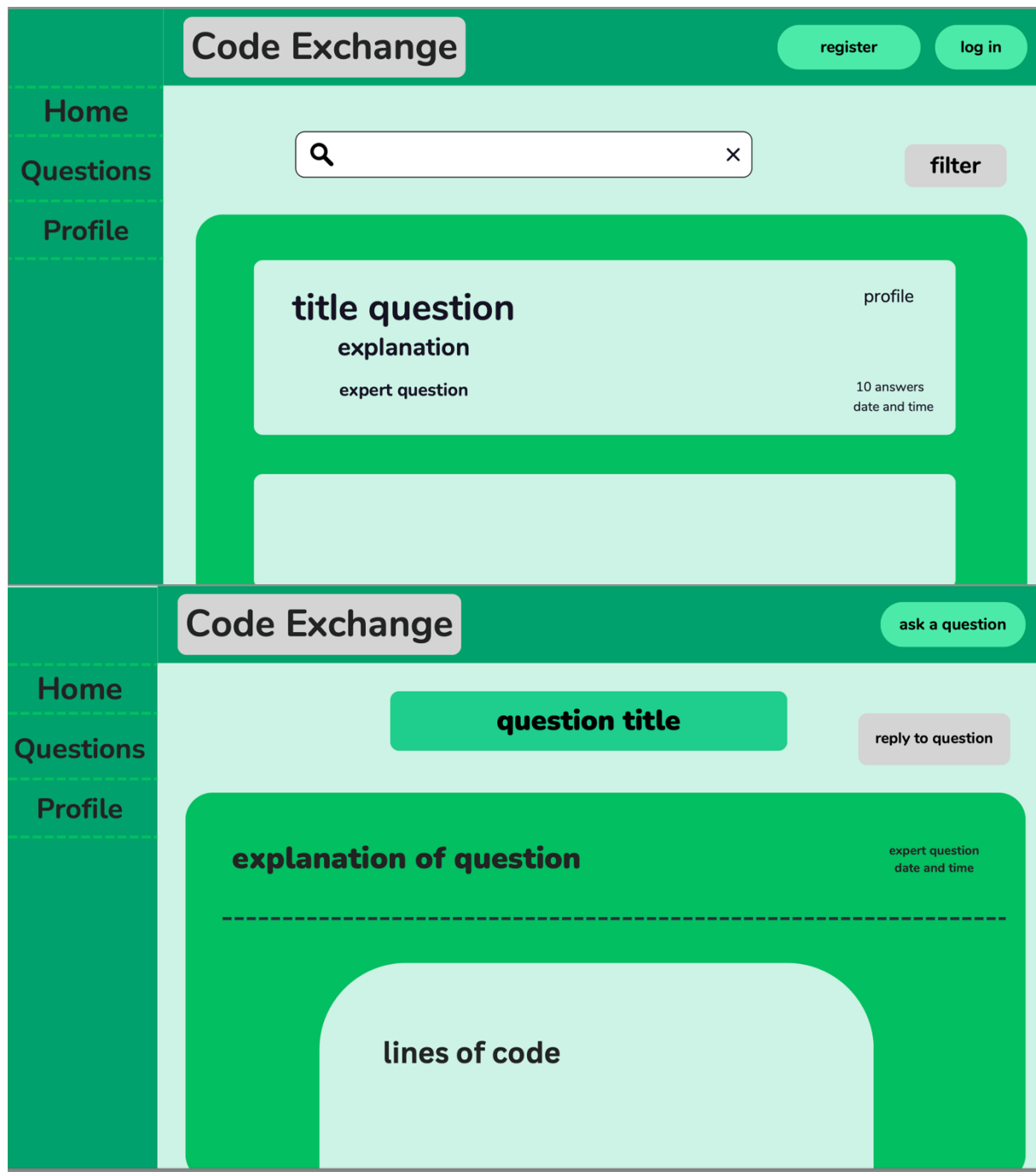
4. CRITERIA

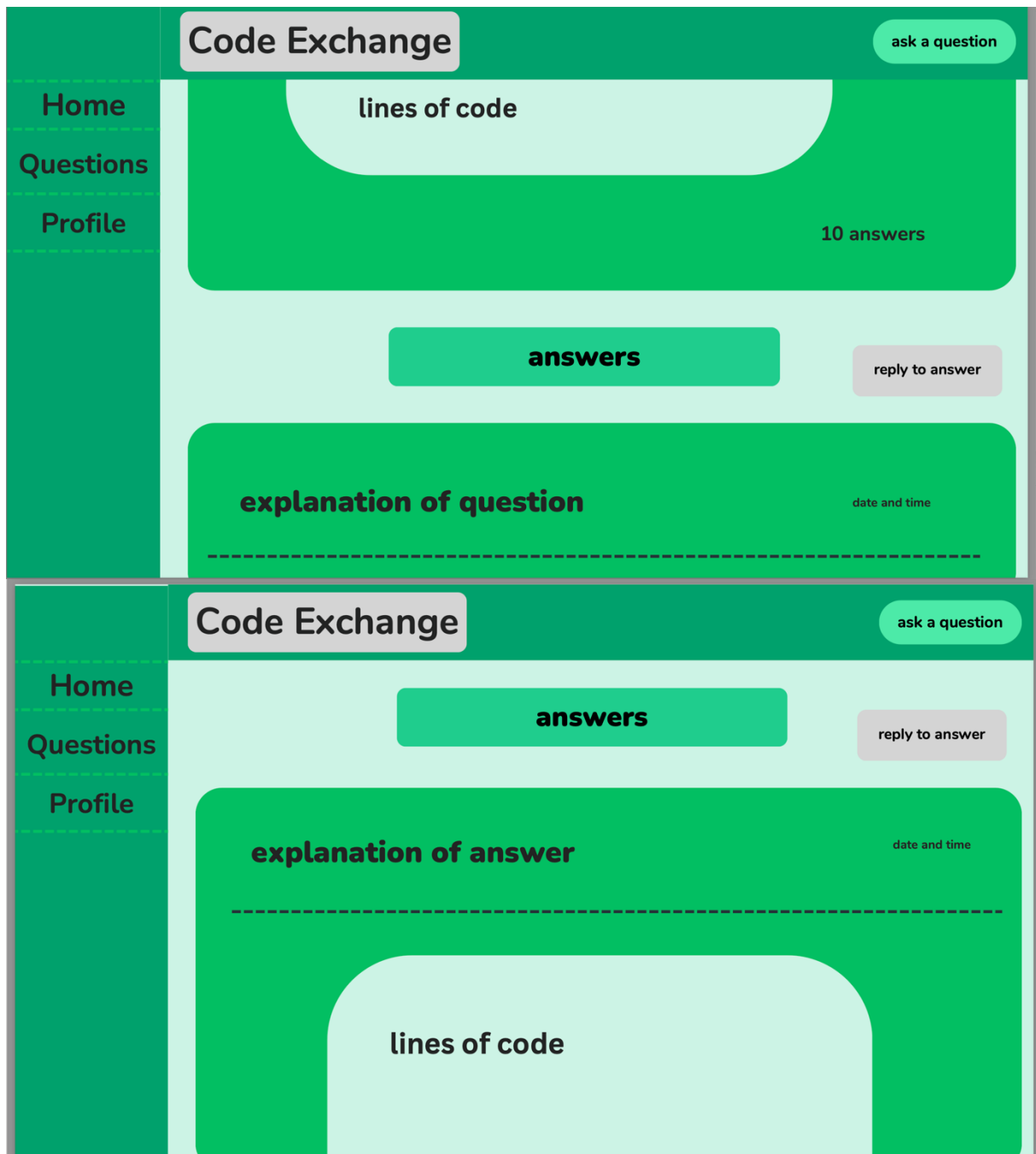
De hypothese moet gevalideerd worden als...

De hypothese moet verworpen worden als...

Het registreren en question maken in een keer lukt

Het met moeite kunnen registreren of questions maken.





Startt learningstory

Ik wil aantonen dat ik werk volgens kwaliteitscriteria.

Situatie: Ik moet kunnen aantonen dat ik werk volgens de kwaliteitscriteria.

Taak: Ik moet aantonen in mijn code dat ik werk met coding conventions, juist manier documenteer, professioneel schriftelijk communiceer, gestructureerd documenten schrijf en vooraf nadenken wanneer een userstory aan eisen voldoet.

Aanpak: Voor coding conventions moet mijn code aan een paar eisen voldoen. Ik had al coding conventions goed aangepast tijdens het maken van mijn code. Dus ik heb niks veranderd eraan. Voor technisch documenteren moest ik ook niks veranderen, want al mijn documenten hebben goed taalgebruik. Ik moest voor professioneel schriftelijk documentatie naar de knowledgebase, zodat ik een goede basis heb. Daarna had ik het toegepast in mijn documentatie. Voor gestructureerd documentatie schrijven moest ik opzoeken op de internet en toepassen op mijn documenten. Bij vooraf nadenken over wanneer een userstory aan eisen voldoet is het vanzelfsprekend. Ik keek naar een userstory en check mark alles die af is. Daarna naar DoD kijken.

Resultaat: Ik kon coding conventions toepassen in mijn code. Ik kan technisch documenteren, professioneel schriftelijk communiceren, gestructureerd schrijven en ik kan vooraf nadenken wanneer een userstory aan eisen voldoet.

Reflectie: Ik vond coding conventions gebruiken makkelijk, het ging ook goed. Technisch documenteren ging moeilijker. Daar moest ik meer op inzetten. Ook bij professioneel communiceren en gestructureerd schrijven ging het moeilijker. Voor af nadenken over een userstory ging makkelijk, want je hoort alleen naar de checkmarks te kijken onder een userstory.

Transfer: Ik wil wat ik geleerd heb over documentatie meenemen naar het volgende blok.

Start met learningstory

Ik wil aantonen dat ik gestructureerd en begrijpelijke documentatie schrijf.

Situatie: Ik moet kunnen aantonen dat ik gestructureerd en begrijpelijk documentatie schrijf.

Taak: Ik moet kunnen actief schrijven, bijna foutloos schrijven volgens de criteria van de stijlguides. Ook moet ik gestructureerd documenteren met alinea's, titels en tussenkoppen.

Aanpak: Ik ging in de knowledgebase opzoeken hoe je moet documenteren. Ook heb ik naar de stijlguides moeten kijken. Dan heb ik het geïmplementeerd in mijn documenten.

Resultaat: Ik heb nu documenten die alinea's hebben, titels en tussenkoppen. Ook heb ik geprobeerd om volgens de criteria foutloos te schrijven.

Reflectie: Ik moest mijn documenten toepassen met alinea's, titels en tussenkoppen. Dat vond ik makkelijk gaan. Het foutloos schrijven vond ik moeilijk. Ik ben namelijk niet zo goed met spelling en grammatica. Dat ging dus moeizaam.

Transfer: Ik wil alles wat ik heb geleerd over foutloos schrijven en gestructureerd documenteren meenemen naar het volgende blok. Ook wil ik mijzelf verbeteren in de toekomst.

Startt met learningstory

Ik wil leren hoe ik een userstory schrijf op basis van de wensen van de gebruiker

Situatie: Ik moet leren hoe ik een userstory maak op basis van de gebruiker.

Taak: Ik moet een userstory maken met een concreet titel, DoD en een acceptatiecriteria. Ik moet gebruik maken van de industriestandaard om userstories te schrijven. Ik moet userstories toevoegen aan het juiste issue bord.

Aanpak: Ik moest een gebruikerstest uitvoeren. Uit die gebruikerstest kwam uit dat een vraag maken niet efficiënt was. Dus het kostte veel clicks voordat je een vraag kon posten. Ik had dan een userstory gemaakt: Als gebruiker wil ik zo efficiënt mogelijk een vraag kunnen stellen.

Resultaat: Ik heb een userstory gemaakt volgens de wensen van een gebruiker.

Reflectie: Ik moest eerst een gebruikerstest doen voordat ik een userstory kon maken. Een userstory maken is niet moeilijk. Het maken van een acceptatiecriteria ging moeilijk, want ik wist niet hoe ik de userstory moest editen. Uiteindelijk is het gelukt.

Transfer: Ik wil het schrijven van userstories meenemen naar het volgende blok.