

Leeruitkomst 1: Je baseert je keuzes op feedback van stakeholders en onderbouwt ze op een heldere en professionele wijze.

### **Stakeholder**

Iemand met een bepaalde rol en belang in het project (ongeacht of deze groot of klein is). Je kunt deze stakeholders identificeren en kunt hun belangen achterhalen en prioriteren.

### **Helder en professioneel**

Documentatie is compleet maar compact, niet-triviaal, gecontroleerd op spelfouten en toepasselijk voor de stakeholder waar deze voor bedoeld is.

### **Onderbouwen**

Geef betrouwbare en relevante bronnen voor alle beslissingen. Je beoordeelt bronnen op hun betrouwbaarheid en relevantie voor het project.

Reflectie:

Omdat ik zelf het project ha kiezen ben ik natuurlijk een van de belangrijke personen in dit project en zal ik ook mijn eisen hierin opschrijven. Daarnaast ga ik het beste deze eisen omzetten in een plan waar ik mijn keuze verantwoord en deze met de docent doorneem.

Leeruitkomst 2: Je werkt samen en communiceert met anderen op constructieve en professionele wijze.

**Professioneel samenwerken.**

Je werkt samen aan een gemeenschappelijk doel en neemt initiatieven om het proces te verbeteren.

**Professioneel communiceren.**

Je levert artefacten op aan de stakeholders en hebt zinvolle meetings met het team. Een artefact is een opgeleverd (deel)product dat waarde heeft voor de stakeholder. Voorbeelden zijn: analysedocumenten, ontwerpdocument, code en geïnstalleerde software.

**Constructief**

Je reflecteert regelmatig op de manier waarop je werkt en hoe je handelen jou, anderen en het projectresultaat beïnvloedt. Hiervoor vraag je regelmatig feedback. Op basis van de informatie die je hieruit verkrijgt, maak je aanpassingen aan je gedrag.

Reflectie:

Ik blijf in contact met mijn docenten en zal minimaal 2 keer in de week feedback vragen over waar ik mee bezig ben daarnaast zal ik hun feedback noteren en toepassen waar dat nodig is.

Leeruitkomst 3: Je documenteert gevalideerde gebruikersspecificaties voor applicaties en vertaalt deze in correcte softwareontwerpen.

### **Gevalideerde gebruikersspecificaties**

Gevalideerd: je controleert dat requirements geaccepteerd zijn door de stakeholders en kunt ze zodanig prioriteren dat de eisen die de meeste waarde opleveren voor de stakeholders, de hoogste prioriteit krijgen.

### **Gebruikersspecificaties**

Het verwachte gedrag van het systeem, gespecificeerd in termen van interactie tussen de gebruiker en het systeem. Specificaties worden gevalideerd met behulp van uitvoerbare acceptatietests.

### **Correcte softwareontwerpen**

Je vertaalt specificaties naar relevante diagrammen waarin het technisch ontwerp beschreven wordt en die kunnen worden geïmplementeerd. Diagrammen die relevant kunnen zijn voor het implementeren van het product zijn onder andere: architectuurdiagrammen, domeinmodellen en databaseontwerpen.

Reflectie:

Ik maak alle benodigde documentatie dat bij een project hoort zoals een functioneel ontwerp en een technisch ontwerp, Dit heb ik al vaak genoeg gedaan op mijn vorige opleiding dus zal hier geen problemen mee hebben

Leeruitkomst 4: Je bouwt, ontwerpt en levert herhaaldelijk veilige en onderhoudbare applicaties op (waarvan er tenminste één web-gebaseerd is) die verbinding maken met een database en gebruik maken van OO-principes en standaard technieken gebaseerd op gevalideerde gebruikerseisen.

### **Herhaaldelijk**

Je maakt, breidt uit en onderhoud verschillende projecten.

### **Ontwerpen**

Het uitbreiden en onderhouden van projecten begint met herijken van de specificaties en het ontwerp.

### **Opleveren**

Je stelt de software op een dusdanige wijze beschikbaar dat de stakeholders er gebruik van kunnen maken.

### **Veilig**

Een softwaresysteem moet bescherm zijn tegen onbedoelde of onverwachte fouten. Onbedoelde fouten treden op als gebruikers het product gebruiken op een manier die niet was voorzien bij het opstellen van de specificaties. Onverwachte fouten treden op als iets faalt in het systeem, zoals bijvoorbeeld het niet beschikbaar zijn van een verbinding met de database.

### **Onderhoudbaar**

Een ontwerp moet klaar zijn voor toekomstige nieuwe eisen of aanpassingen aan bestaande specificaties.

### **OO Principes**

Gebruik gangbare OO-principes om het softwaresysteem te ontwerpen en dit ontwerp te onderbouwen.

### **Reflectie:**

Het project zal gemaakt worden naar aanleiding van de documentatie die ik maak zal uitgebreid worden waar nodig is en als ik eerder klaar mocht zijn dat ik nog een ander klein project zal doen. Gebruikers data zal veilig opgeslagen door middel van encryptie.

Leeruitkomst 5: Je redeneert over computationele uitdagingen en implementeert algoritmisch complexe problemen in software.

### **Computationele Uitdagingen**

Je lost veelvoorkomende uitdagingen op, zoals een sorteerprobleem en onderbouwt waarom een bepaalde techniek geschikt is voor specifieke problemen.

### **Algoritmisch complexe problemen**

Je ontwerpt en implementeert algoritmes zodanig dat ze rekening houden met de randvoorwaarden van de stakeholders.

Reflectie:

Er zal een stuk complexiteit in komen te zitten voor het genereren van de user databases en natuurlijk in het gameplay stuk

Leeruitkomst 6: Je ontwerpt, bouwt en bevroagt een relationeel databasesysteem en integreert dit met een applicatie.

### **Ontwerpen**

Maak onderscheid tussen database- en softwareontwerpen. Een databaseontwerp bevat meerdere soorten relaties, zoals 1-op-veel en veel-op-veel.

### **Bevragen**

Je voert CRUD-operaties (Create, Read, Update en Delete) uit op data in een relationele database, en houdt rekening met performantie.

Reflectie:

Omdat de game volledig afhankelijk zal zijn van de database die er mee verbonden is. En het zal de standaard CRUD operaties bevatten

Leeruitkomst 7: Je verbetert en toont de kwaliteit van je software continue aan, gebruikmakend van standaard technieken en hulpmiddelen.

### **Continue**

Je werkt op een iteratieve wijze zonder bestaande functionaliteit te verstoren en waarbij veranderingen worden bijgehouden.

### **Verbeteren**

Gebruik standaard hulpmiddelen en technieken om de kwaliteit van je code te bewaken en te verbeteren.

### **Aantonen**

De code moet getest worden voor zowel het geplande gebruik, verwachte en onverwachte foutsituaties. Deze testen moeten meerdere malen kunnen worden uitgevoerd in meerdere fasen van het project. Verwachte fouten kunnen voortkomen uit de specificaties of van externe afhankelijkheden van de software.

### **Standaard technieken en hulpmiddelen.**

Gebruik bijvoorbeeld een versiebeheersysteem, acceptatietests en unit-tests.

### **Reflectie:**

Iedere sprint werk ik om een deel van de applicatie op te leveren en stel deze doelen aan het begin van de sprint. Ook zal mijn code getest worden door een simpel test plan te maken