

UML-ONTWERP & -REALISATIE

Naam	Anthony Delgado
Studentnummer	22102167
Klas	H-1-SE.8

OPDRACHT

Geef hieronder een korte beschrijving van je opdracht (een samenvatting van je productvisie voor het vak H-SE-S2-REQS) zodat je docent OPT2 ook begrijpt met welke opdracht jij bezig bent:

<Opdracht>

Mijn opdrachtgever wilt een applicatie waarmee hij de verandering van de prijs van een gegeven product kan volgen. Dit wilt hij door kunnen doen door producten aan te maken, een categorie toe te wijzen en bij een product in te kunnen vullen op welke dag hij het product voor welke prijs heeft gekocht. Uiteindelijk is het de bedoeling dat dit weergegeven wordt in een timeline als een lijngrafiek.

Mijn opdrachtgever wilt ook kunnen budgetten met de applicatie. Dit wilt hij kunnen doen door categorieën aan te maken, en vervolgens voor een gegeven categorie een product met prijs, beschrijving en datum in te voeren. Hij wilt dat hij een tijdsperiode kan kiezen om deze gegevens in verschillende diagrammen te laten weergegeven, zoals een kolom- of cirkeldiagram.

GITHUB

Maak je GitHub-repo private en geef je docent voor het practicum toegang tot je repo (vraag hem om zijn account als hij dat volgens jou nog niet aan jou gegeven heeft):

<Link naar GitHub-repo>

https://github.com/DKnightAnon/SE1_Semester_opdracht

UML-ONTWERP

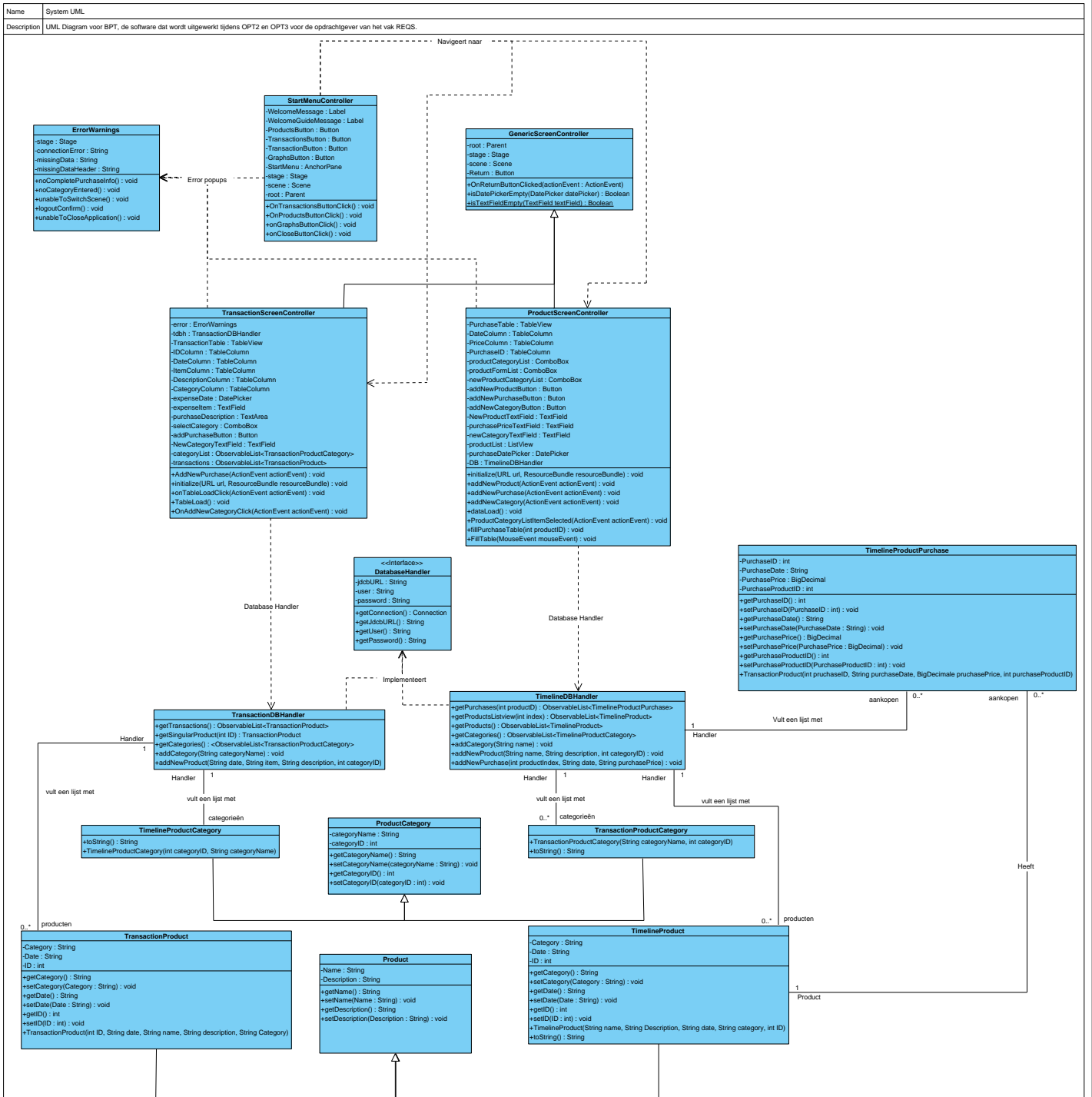
Hieronder heb ik een kopie gemaakt van mijn ontwerp. Het is verleidelijk om hier een ingewikkeld model te laten zien, maar dat hoeft niet. Zie het volgende document voor voorbeelduitwerking en aanwijzingen:

[OPT2 - Portfolio - UML-Classmodel en -realisatie - Toelichting.pdf](#)

Onderstaand ontwerp heb ik opgesteld om de volgende User Stories te kunnen realiseren :

1	'Als gebruiker wil ik voor producten een aanmaak formulier hebben, zodat ik producten kan aanmaken en data(bijvoorbeeld prijs op een gegeven datum) van een product kan invoeren.'
2	'Als <Gebruiker van de software> wil ik <Categorieën kunnen aanmaken en toewijzen aan producten>, zodat ik <Per categorie kan zien hoeveel ik eraan besteed.
3	'Als Gebruiker van de software wil ik per product elke keer kunnen invoeren wanneer ik het heb gekocht en voor welk bedrag, zodat ik <Een overzichtelijke lijst heb voor de verandering van de prijs.
4	'Als gebruiker van de software wil ik dat alle naar alle tabellen kan navigeren vanuit hetzelfde scherm, zodat ik Alle informatie in een plek heb, maar het toch gescheiden is.'
5	'Als <Gebruiker> wil ik <Apart producten kunnen invoeren waarvan ik de prijs wil volgen, en producten dat ik (een enkele keer) gekocht heb>, zodat ik specifieke producten met een timeline heb en productaankopen waarvan ik in een grafiek kan zien hoeveel ik besteed per categorie.'

User Story 5 is gedeeltelijk uitgewerkt. In plaats van grafieken bevatten de pagina's momenteel tabellen.



<Copy-Paste hier een kopie van je domeinmodel>

* Controleer of het plaatje goed leesbaar is.

Hieronder som je voor elke *UML Class* in je ontwerp op wat de functionaliteit is van die *Class* (voor zover je dat nu al kunt weten) en voor elke associatie in het *UML Class Diagram* wat de toegevoegde waarde is van die associatie:

Naam van de Class	Functionaliteit van genoemde Classes
ErrorWarnings	Deze class bevat methodes die aangeroepen worden vanuit andere classes om zo foutmeldingen op het scherm te weergeven.
GenericScreenController	Deze class is de basis van alle ScreenControllers behalve de StartMenuController. Hierin zitten een paar basis methodes zoals een knop om terug naar het startmenu te gaan, en methodes om te kijken of bepaalde velden in de andere schermen een waarde hebben of niet.
TransactionScreenController	Deze class regelt het gedrag van het Transaction scherm. Hier kun je nieuwe categorieën aanmaken en producten invoeren in de database. Om een product aan te maken moet je ook een categorie aanwijzen. Deze class zorgt ervoor dat alle input in de aanmaak formulieren naar de TransactionDBHandler class wordt gestuurd, en verzamelt de output van diezelfde class en toont het in tabellen op het scherm.
ProductScreenController	Deze class regelt het gedrag van het Product scherm. Hier kun je producten invoeren en prijzen met aankoopdatums voor geselecteerde producten doorgeven. Deze class zorgt ervoor dat alle input in de aanmaak formulieren naar de TimelineDBHandler class wordt gestuurd, en verzamelt de output van diezelfde class en toont het in tabellen op het scherm.
ProductCategory	Deze class is de basis van alle andere <i>-ProductCategory</i> classes. Deze class bevat een paar basis attributen.
TransactionProductCategory	Deze class wordt door TransactionDBHandler gebruikt om een lijst met categorieën op te halen uit de database. Elke categorie wordt in een nieuw object gezet van deze class. Deze categorieën zijn specifieke categorieën die voor transactieproducten zijn aangemaakt, ofwel Grafiekproducten.(Zie User Story 5)
TimelineProductCategory	Deze class wordt door TimelineDBHandler gebruikt om een lijst met categorieën op te halen uit de database. Elke categorie wordt in een nieuw object gezet van deze class. Deze categorieën zijn specifieke categorieën die voor Timelineproducten zijn aangemaakt.(Zie User Story 5)
Product	Deze class is de basis van alle andere <i>-Product</i> classes. Deze class bevat een paar basis attributen.
TransactionProduct	Deze class wordt door TransactionDBHandler gebruikt om een lijst met producten op te halen uit de database. Deze producten zijn specifiek transactieproducten, ofwel Grafiekproducten .(Zie User Story 5)
TimelineProduct	Deze class wordt door TransactionDBHandler gebruikt om een lijst met producten op te halen uit de database. Dit zijn specifiek Timelineproducten .(Zie User Story 5)
DatabaseHandler	Dit is een interface dat geïmplementeerd wordt door alle <i>-DBHandler</i> classes. Hierin zitten een paar static variabelen waarmee de verbinding met de database wordt geregeld.

TransactionDBHandler	Deze class krijgt de input vanuit TransactionScreenController en regelt het opvragen en invoeren van data in de database.
TimelineDBHandler	Deze class krijgt de input vanuit ProductScreenController en regelt het opvragen en invoeren van data in de database.
TimelineProductPurchase	Deze class is specifiek voor Timelineproducts. Deze class wordt gebruikt door TimelineDBHandler om een lijst met aankopen van een specifiek product op te halen uit de database.
<Class>	<functionaliteit van de class>

Voor bovenstaande classes geldt het voor de Controller classes dat ze gebruikmaken van een .fxml bestand.

Naam van de associatie	Toegevoegde waarde van de genoemde associaties
Error Popups	De screencontrollers kunnen gebruik maken van de methodes in ErrorWarnings om foutmeldingen te weergeven als er iets fout gaat of verkeerde input ingevoerd wordt.
Database Handler	Als een gebruiker informatie wilt opvragen of invoeren in de database, moet deze informatie verwerkt worden.
“Navigeert naar”	Een gebruiker moet kunnen schakelen tussen verschillende schermen.
<Associatie>	<toegevoegde waarde van de associatie>
<Associatie>	<toegevoegde waarde van de associatie>
<Associatie>	<toegevoegde waarde van de associatie>

REALISATIE

De eerste versie van mijn software die ik heb gebaseerd op het ontwerp:

Zie de GitHub link.

https://github.com/DKnightAnon/SE1_Semester_opdracht