**EINDONTWERP**

|  |  |
| --- | --- |
| **Naam** | Xue Kun Zhou |
| **Studentnummer** | 20170173 |
| **Klas** | 6 |

**OPDRACHT**

Geef hieronder een korte beschrijving van je opdracht (een samenvatting van je productvisie) zodat je docent OPT3 ook begrijpt met welke opdracht jij bezig bent (neem dit onderdeel over uit je laatst ingeleverde opdracht en pas het aan aan de hand van je eigen bevindingen en aan de hand van feedback van medestudenten en je docent aan):

MyLibrary is een webapp dat bijhoudt welke films, series, boeken en games je hebt gezien, gelezen en gespeeld. De gebruiker heeft daarmee een digitale “collectie” van media die hij heeft geconsumeerd. Per item wordt er ook een score bijgehouden zodat wanneer de gebruiker op een later moment kan terugzien wat hij ervan vond. Door het bijhouden van deze items kan de gebruiker ook een overzicht krijgen van hoeveel tijd hij hierin besteed.

De app houdt niet bij welke films etc. er al bestaan en houdt dus geen gemiddelde score bij van alle gebruikers en geeft ook geen aanbevelingen van welke items de gebruiker misschien wel leuk vindt.

Vrienden kunnen wel de lijsten van elkaar zien.

**GITHUB**

Maak je GitHub-repo Private en geef je docent voor het werkcollege toegang tot je repo (vraag hem om zijn account als hij dat volgens jou nog niet aan jou gegeven heeft). Stuur niet steeds een nieuwe link/uitnodiging, maar bouw in feature-branches verder aan de applicatie die je eerder hebt opgeleverd:

https://github.com/XuekunZhou/OPT3\_MyLibrary (in branch main)

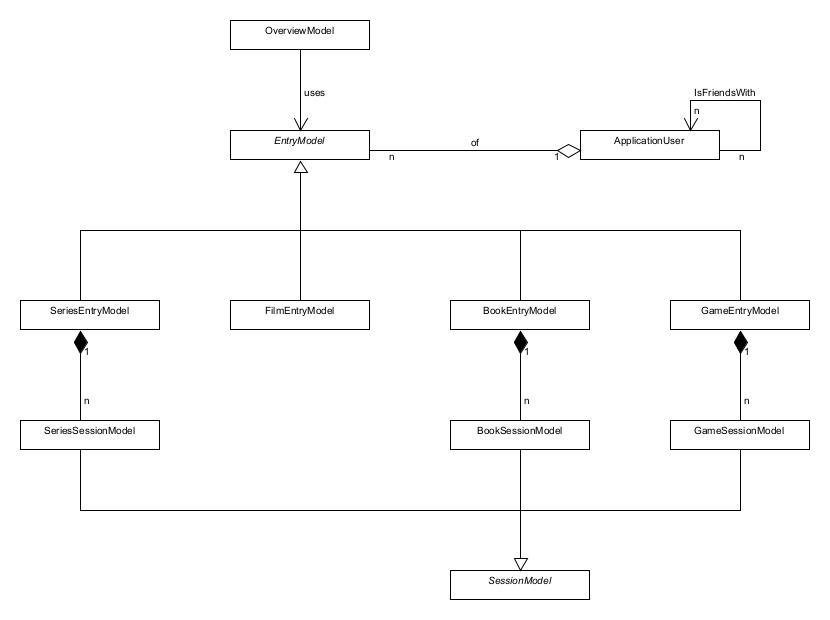
Bovenstaande repo is m.b.v. .gitignore ontdaan van IntelliJ-files en er is een Maven POM-file toegevoegd waarmee je applicatie in één keer gerund kan worden.

**DOMEINMODEL**

Bij de oplevering van het ontwerp en de realisatie hieronder ben ik uitgegaan van de volgende User Stories (gebruik de ‘+’ rechts van de tabel om een User Story toe te voegen):

|  |
| --- |
| Als Gebruiker wil ik film, series, etc. een score geven zodat ik kan bijhouden welke films, series, etc. ik gezien heb en wat ik er van vond. |

Kopieer hieronder (een kopie van) het (aangepaste) domeinmodel dat je als resultaat hebt opgeleverd bij de vorige opdracht (kopieer het model uit je vorige opdracht of kopieer het aangepaste model):

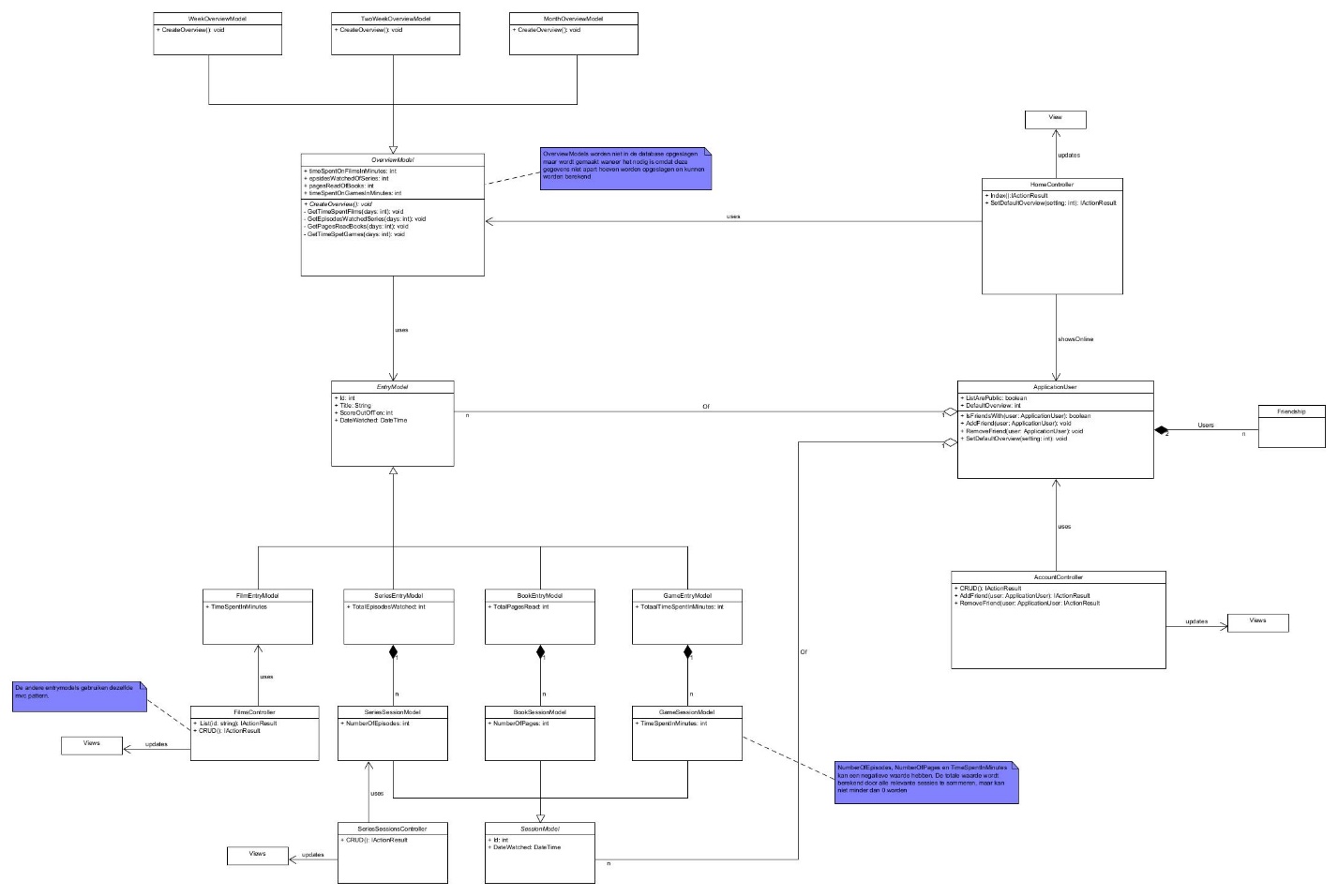


\* Controleer of het plaatje goed leesbaar is. Als je wilt dat je docent het domeinmodel met realisatie opnieuw beoordeelt, lever je het ingevulde template voor het domeinmodel in via de inleverbox voor de herkansing van het domeinmodel in BlackBoard.

**EINDONTWERP**

Vanuit dit domeinmodel heb je Patterns en SOLID-principes toegepast op je model en je hebt Classes toegevoegd die je nodig hebt om je applicatie te laten werken (zoals bijv. voor het tonen van een GUI of Command Line voor het tonen en afhandelen van een menu). Je hebt er kortom een technisch ontwerp van gemaakt waarin je ook de types van je properties, parameters en methodes hebt toegevoegd. Zie het volgende document voor voorbeelduitwerking en aanwijzingen:

‘[OPT3 16.1.3.2 - Eindontwerp – Toelichting.docx](https://blackboard.hhs.nl/bbcswebdav/courses/H-SE-S2-AOOP-2020/OPT3/OPT3%2014.1.3.2%20-%20Domeinontwerp%20en%20-realisatie%20-%20Toelichting.pdf)’



\* Controleer of het plaatje goed leesbaar is.

**PATTERNS**

Beschrijf hieronder hoe je het Template Method Pattern en een tweede Pattern hebt toegepast in bovenstaand eindontwerp:

|  |  |
| --- | --- |
| **Pattern** | **Beschrijving** |
| Template Method Pattern | Na(a)m(en) van template method (s): SetTimeSpentOnFilms(), SetEpisodesWatchedOfSeries(), SetPagesReadOfBooks(), SetTimeSentOnGames()  Afhankelijk van de method wordt er uit een groter periode data verzameld |
| Observer Pattern | Model-View-Controller pattern maakt gebruik van de observer pattern |

**SOLID-PRINCIPES**

Beschrijf hieronder voor twee SOLID-principes hoe je die hebt toegepast in Bovenstaand eindontwerp:

|  |  |
| --- | --- |
| **SOLID-principe** | **Beschrijving (gebruik een spatie als je geen toelichting geeft)** |
| Single Responsibility | Een Controller maakt alleen gebruik van 1 Model |
| Open-Closed | EntryModel is abstract. Als er in de programma nieuwe type EntryModel word toegevoegd kan je de EntryModel extenden en hoeft hij niet worden bijgewerkt |
| Liskov | HomeController kan gebruik maken van alle subclasses van OverviewModel zonder raar gedrag tegen te komen |
| Interface Segragation |  |
| Dependency-Inversion |  |

**REALISATIE**

Als laatste onderdeel voor deze opdracht kopieer je hieronder de tweede versie van de Javacode waarmee je het eindontwerp realiseert (hoewel je de code smells al wel uit deze code mag verwijderen wordt dat pas beoordeeld bij de eindoplevering):

<De code van de software op basis van bovenstaand domeinontwerp>