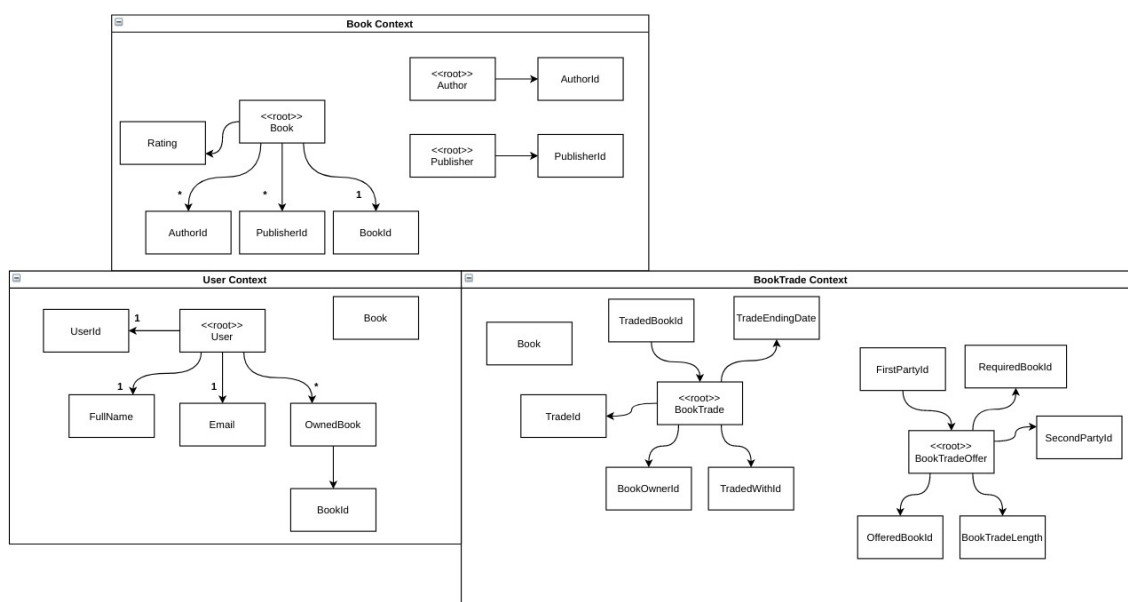


Електронска и мобилна трговија 2020 - Изработка на проект за лаб. вежби - Фаза 1	
Индекс, име и презиме на членовите од тимот	1. 171542, Егзон Исеини 2. 172003, Агон Џафери
Наслов на проектот	Book Trading

Дефинирајте сценарио треба да го имплементирате со користење на алатките од Domain Driven Design и кое треба да биде по сложеност слично на сценариото на аудиториски вежби. Како потсетување, користете ги презентациите за Strategic Design и Tactical Design кои се поставени на курсот. Во фаза 2 истото сценарио ќе треба да го испрограмирате.

1. Опишете го сценариото кое го обработувате (200 збора).



Целта на апликацијата односно платформата е да се овозможи на корисниците и на сите љубители на книгата да можат да разменуваат книги меѓусебе.

Секој корисник при регистрација и најава, може да вметнува книги во неговиот профил кој тој ги поседува и сака да ги размени со други книги. Книгите би се зимале од надворешен source односно некое API, но би се чувале и во наша база. При додавање на тие книги, корисникот дава до знаење дека тие книги ги поседува и дека сака да ги размени со други книги.

Постои централно место каде корисниците можат да ги најдат сите книги кој се поставени за размена. За секоја книга би се чувале информации како авторите на книгата, рејтингот на книгата, издавачката куќа на книгата итн. Воедно за книгата би се покажале и сите корисници кој што ја поседуваат таа книга.

Потоа еден корисник може да прати понуда кон друг корисник (да кажеме странка 1 и странка 2), каде што странка 1 ја побарува книгата што странката 2 ја нуди, и воедно предлага некоја негова книга и времетраење на размената. Странката 2 може да ја прими понудата или да даде контра предлог, пример да го промени времето на размена или да побара некоја друга книга странката 1. Доколку не се успеат да договората, понудата паѓа и не се случува размена.

Системот реално би бил тешко изводлив и комплициран, но веруваме дека би бил одличен прв ДДД проект за нас и ќе научиме доста ДДД концепти од истиот.

2. Идентификувајте ги ентитетите и релациите меѓу нив кои ви се потребни за имплементација на сценариото. Бројот на дефинирани ентитети мора да биде најмалку 3.

Иницијалните ентитети кои би ги имале во овој систем би биле:

- Book Context (Book, Author, Publisher)
- User Context (User, OwnedBook)
- BookTrade Context (BookTrade, BookTradeOffer)

3. Идентификувајте ги потребните атрибути за секој од ентитетите.

Некој од атрибутите би биле следните (не е финална верзија, ќе претрпи огромни промени):

Book

- > id: BookId
- > bookName: String
- > bookAuthor: Author*
- > publisher: Publisher*
- > rating: Rating

Author

- > id: AuthorId
- > fullName: FullName

Publisher

- > id: PublisherId
- > publisherName: String

BookTrade

- > id: TradeId
- > BookOwnerId: UserId

- > TradedWithId: UserId
- > TradedBookId: BookId
- > TradeEndingDate: LocalDateTime
- > hasTradeEnded: Boolean

BookTradeOffer

- > FirstPartyId: UserId
- > SecondPartyId: UserId
- > RequiredBookId: BookId
- > OfferedBookId: BookId
- > BookTradeLength: LocalDateTime
- > hasFirstPartyAccepted: Boolean
- > hasSecondPartyAccepted: Boolean
- > hasOfferBeenAccepted: Boolean

User

- > id: UserId
- > fullName: FullName
- > emailAddress: Email
- > password: String
- > ownedBooks: OwnedBook*

4. Идентификувајте ги ограничените контексти (bounded contexts) во вашето сценарио.

Како што претходно напоменав, имаме три ограничени контексти:

- Book Bounded Context
- User Bounded Context
- BookTrade Bounded Context

5. Идентификувајте ги агрегатите во секој од ограничените контексти.

- Book Context (Book, Author, Publisher)
- User Context (User)
- BookTrade Context (BookTrade, BookTradeOffer)

6. Идентификувајте го Aggregate Root на секој од идентификуваните агрегати.

- Book Context (Book, Author, Publisher)
- User Context (User)
- BookTrade Context (BookTrade, BookTradeOffer)

7. Идентификувајте неколку правила за конзистентност (бизнис правила) во сценариото. Специфицирајте кој ентитет ќе ги поседува имплементациите на истите?

Некој од бизнис правилата би биле (ќе претрпат измени најверојатно):

1. Корисник мора да ја поседува книгата пред да ја понуди.
2. Кога ќе истече рокот за размена на книгата се известуваат корисниците.
3. Веќе постоечки разменето OwnedBook на корисник неможе да се постави за оглас или да се понуди пак за размена пред да заврши инцијалната размена.
4. BookTradeOffer се води како успешна само доколку двете странки го потврдат тоа итн.

8. Идентификувајте неколку вредносни објекти (value-objects) во вашето сценарио. Кои методи би ги имплементирале?

Оваа сценарио има доста вредносни објекти, како што се на пример:

- Shared Kernel (FullName, Email, BookId, UserId, Rating)
- BookTradeLength, TradeEndingDate, FirstPartyId, SecondPartyId etc.

9. Кои се предностите при користење на вредносни објекти (value-objects)?

Имутабилноста на истите многу ги поедноставува работите. Воедно тие овозможуваат да имаат внатрешна бизнис логика.

10. Идентификувајте неколку настани (events) кои треба да протекуваат помеѓу агрегатите.

Кога BookTradeOffer ќе биде потврдено, се праќа евент до User Context каде книгите кои се потврдени за размена им се менува availability.

Воедно при завршување на deadline се испраќаат евенти кон корисниците и ги информираат за тоа.