

Проектна задача по Електронска и мобилна трговија (2021/2022)

3D Manager

Виктор Мегленовски – 191001

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

3D Manager е апликација наменета менаџирање на проекти на фриленсери кои се занимаваат со 3D моделирање и рендерирање. Самата апликација овозможува фриленсерите да се поврзат со клиентите. При активирање на проект помеѓу фриленсерот и клиентот ќе се генерираат низа од активности кои ќе овозможат корисникот да го прикажува и ажурира текот на проектот (нешто слично како во агилните методи за програмирање), а воедно и клиентот ќе може да го следи целиот процес на работа на фриленсерот.

Апликацијата нема да овозможува систем за наплата и трансфер на средства помеѓу клиентите и фриленсерите, туку ќе служи како помошен систем за подобра и поефективна координација на проектите.

Фриленсерите ќе можат да прикачуваат и додаваат задачи на секој проект, да го менуваат статусот на задачите, да пуштаат нотификации кон клиентите и слични активности, додека пак клиентите ќе можат да го следат напредокот на проектот, да оставаат коментари на задачите, да потврдуваат дали одредена задача е добро завршена, да побараат ревизија на одредена задача и слично.

Ова сценарио е погодно за дизајнирање со DDD затоа што има изобилство на бизнис процеси кои многу поприродно и подобро ќе бидат имплементирани со алатките од DDD отколку со стандардна CRUD апликација.

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. Нацртајте го и ЕА дијаграмот и прикачете го како слика.

AppPerson што е генерички термин од кој наследуваат Client и Freelancer.

Project што го претставува јадрото на системот.

Task што претставува задача поврзана со некој проект. Секој Project има повеќе Tasks. Еден Task припаѓа на само еден проект.

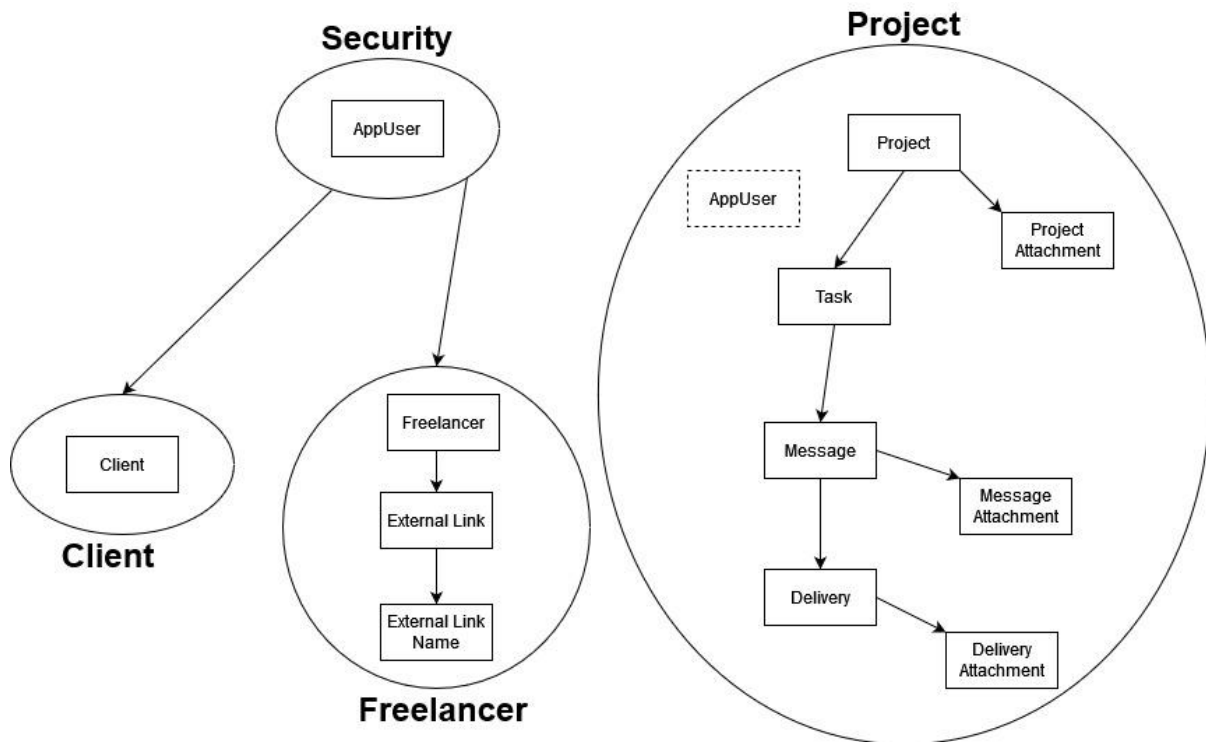
Message што претставува коментар поврзан со дадена задача. Секој Task има повеќе Messages. Еден Message припаѓа на само еден Task.

Delivery што претставува достава поврзана со дадена задача. Секој Task има повеќе Deliveries. Еден Delivery припаѓа на само еден Task.

ExternalLink што претставува линк до профил на друг медиум на фриленсерот (на пример Upwork, LinkedIn, Fiverr и слично). Секој Freelancer има повеќе External Links, секој External Link припаѓа на точно еден Freelancer.

Покрај овие главни ентитети, системот содржи и други ентитети кои претставуваат прикачени фајлови на Project, Message и Delivery ентитетите.

Секој Project има повеќе Project Attachments. Слично, секој Message и Delivery има повеќе Message Attachments и Delivery Attachments соодветно. Еден Attachment припаѓа на еден Project, Message, Delivery соодветно.



3. Идентификувајте ги потребните атрибути за секој од ентитетите. За секој од атрибутите дефинирајте го типот.

AppUser(email:string, name:string, surname:string, password:string, role:enumeration)

Client(email:string, name:string, surname:string, company:string, rating:Rating(mapped as integer), role enumeration)

Freelancer(email:string, name:string, surname:string, rating:Rating(mapped as integer), role enumeration)

ExternalLink(id:string, link:string, name:ExternalLinkName, freelancer:Freelancer)

ExternalLinkName(name:string, imageLocation:string)

Project(id:string, name:string, description:string, dueDate:Date, status:enumeration, folderLocationString, clientRating:int, freelancerRating:int, client:AppUser, freelancer:AppUser)

ProjectAttachment(id:string, project:Project, attachmentPath:string, attachmentName:string)

Task(id:string, title:string, taskStatus:enumeration, project:Project, timestamp:LocalDateTime, folderLocation:string)

Message(id:string, task:Task, entryPoster:AppUser, content:string, postingTime:LocalDateTime, metadata:enumeration)

MessageAttachment(id:string, message:Message, attachmentPath:string, attachmentName:string)

Delivery(id:string, task:Task, content:string, deliveryTime:LocalDateTime, deliveryStatus:enumeration)

DeliveryAttachment(id:string, delivery:Delivery, attachmentPath:string, attachmentName:string)

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

Оваа апликација има 4 ограничени контексти.

Security ограничен контекст којшто е задолжен за: регистрација и најава на корисници, автентикација на корисници, доделување и проверка на JWT токени.

Client ограничен контекст којшто е задолжен за сите активности поврзани со клиентите.

Freelancer ограничен контекст којшто е задолжен за сите активности поврзани со фриленсерите.

Project ограничен контекст којшто е јадрото на целата апликација. Во овој ограничен контекст се извршуваат сите бизнис правила поврзани со креирањето на проекти, креирањето на задачи, пишување пораки, правење доставки, менување на статуси на проекти и задачи, оценување на клиенти и фриленсери и слично.

5. Идентификувајте ги агрегатите во секој од ограничените контексти.

Security ограничениот контекст има само еден агрегат: AppUser.

Client ограничениот контекст има само еден агрегат: Client.

Freelancer ограничениот контекст има два агрегати: Freelancer и External LinkName.

Project ограничениот контекст има еден агрегат: Project.

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

Aggregate Roots на сите горе наведени агрегати се ентитетите со истите имиња како и соодветните агрегати.

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

При потврдување на проект од страна на фриленсерот, статусот на проектот се менува од PROPOSED во IN PROGRESS – имплементирано во ентитетот Project.

При правење на достава на одредена задача, статусот на задачата се менува од IN PROGRESS во DELIVERED – имплементирано во ентитетот Task.

По завршување на сите задачи, клиентот има можност да го заврши целиот проект – имплементирано во ентитетот Project.

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

Вредносен објект Rating којшто се наоѓа во Shared Kernel и се користи во Client и Freelancer модулите. Со овој вредносен објект се чува информација за рејтингот на самиот корисник. При добивање на нов рејтинг, вредноста на постоечкиот рејтинг се менува преку самата логика поставена во вредносниот објект.

Вредносен објект AppUser во Project модулот. Овој вредносен објект се користи за мапирање на корисници од Client и Freelancer модулите во објекти по направен соодветен REST повик. Овој вредносен објект е дел од Project ентитетот во 2 улоги – еднаш за клиентот и еднаш за фриленсерот (во проектот се чуваат email и име на клиентот и фриленсерот).

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

Кога се регистрира нов клиент, Security модулот испраќа настан до Client модулот за да се креира соодветна ставка и во Client модулот.

Кога се прави промена на информациите на клиент, Security модулот испраќа настан до Client модулот за да се направат соодветните промени и во Client модулот.

Кога се регистрира нов фриленсер, Security модулот испраќа настан до Freelancer модулот за да се креира соодветна ставка и во Freelancer модулот.

Кога се прави промена на информациите на фриленсер, Security модулот испраќа настан до Freelancer модулот за да се направат соодветните промени и во Freelancer модулот.

Кога фриленсер остава нов рејтинг за некој клиент (на некој нивен заеднички проект), Project модулот испраќа настан до Client модулот за да се промени рејтингот на клиентот.

Кога клиент остава нов рејтинг за некој фриленсер (на некој нивен заеднички проект), Project модулот испраќа настан до Freelancer модулот за да се промени рејтингот на клиентот.