

Електронска и мобилна трговија 2020 – Изработка на проект за лаб. вежби – Фаза 1	
Индекс, име и презиме на членовите од тимот	1. Ана Ѓорѓевиќ, 161062 2. Катерина Илиевска, 161142
Наслов на проектот	Евиденција на работата на една организација која изработува проекти

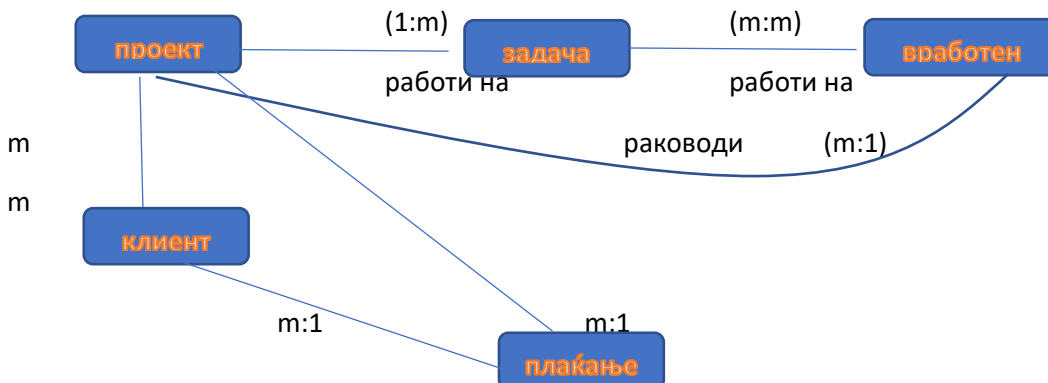
Дефинирајте сценарио треба да го имплементирате со користење на алатките од Domain Driven Design и кое треба да биде по сложеност слично на сценариото на аудиториски вежби. Како потсетување, користете ги презентациите за Strategic Design и Tactical Design кои се поставени на курсот. Во фаза 2 истото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

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

Апликацијата ќе биде наменета за евиденција на работата на некоја организација која изработува некои проекти. Во компанијата се изработуваат различни проекти, за секој проект се евидентираат неговиот наслов, краток опис, deadlines, цена, неговиот менаџер итн. Потоа се чуваат податоци за вработените: име, презиме, телефонски број за контакт, адреса и се евидентираат работните задачи кои тие ги извршиле во рамките на некој проект: опис на задачата, почетен датум, број на часови, сума која треба да му се плати за извршената задача. Се евидентираат и клиентите, притоа еден проект може да биде од еден или повеќе клиенти и на крај се врши евиденција на извршените плаќања од клиентите.

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.

Ентитети се: проект, работна задача кои се поврзани во релација 1:m (еден проект може да има повеќе задачи, а една задача може да се изврши само за еден проект), вработен кој е поврзан со ентитетот работна задача во релација m:m (еден вработен може да изврши повеќе задачи, а една задача може да се изврши од повеќемина), овие два ентитети се поврзани со проект и со релација 1:m (еден проект може да има еден менаџер, а еден вработен може да е менаџер на повеќе проекти), клиент поврзан со ентитетот проект преку m:m релација и плаќање кое е поврзано со клиент и со менаџер со 1:m релација (едно плаќање се однесува на еден проект и е од еден клиент, а за проектот и од клиентот може да има повеќе плаќања).



3. Идентификувајте ги потребните атрибути за секој од ентитетите.

Проект(name, description, start Date, deadline, cost, manager, tasks, clients, finished, payed)

Работна задача(date, description, employees, hours, project\_id, cost)

Вработен (name\_sname, e-mail, address, phone, tariff)

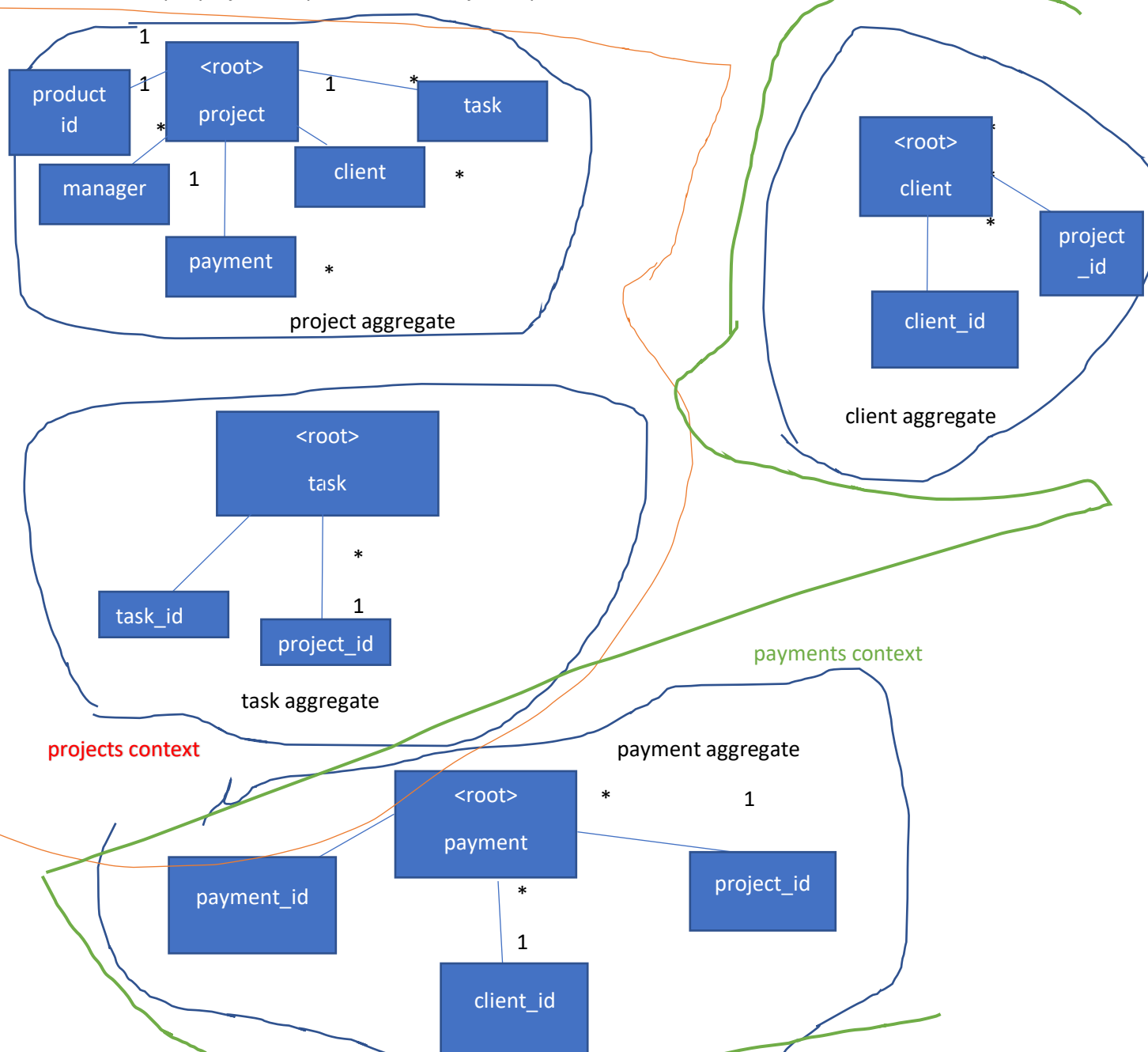
Клиент(name\_sname, e-mail, address, phone, projects, isActive)

Плаќање(date, amount, project\_id, client\_id)

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

Projects bounded context, Payments bounded context

5. Идентификувајте ги агрегатите во секој од ограничените контексти.



Во ограничениот контекст Projects ги идентификувавме агрегатите project aggregate и task aggregate, а во ограничениот контекст Payments агрегатите payment aggregate и client aggregate.

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

Project aggregate – project, Payment aggregate – payment  
(одговорено со цртежот погоре)

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

- кога сите задачи во рамките на еден проект ќе имаат статус finished, проектот добива статус finished, имплементацијата ќе биде од ентитетот кој е aggregate root (project)
- кога целата сума на проектот ќе биде платена проектот добива статус payed, имплементацијата ќе биде од ентитетот кој е aggregate root (project)
- клиентите што имаат некој проект кој сè уште не е payed ќе имаат вредност true на атрибутот isActive, имплементацијата ќе биде од ентитетот кој е aggregate root (client)

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

money (add, subtract multiply, convert), date (get\_hours, get\_minutes, get\_year, get\_month, get\_day), address (get\_street, get\_city, get\_country).

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

Полесно разбирање на работите од страна на лицата кој се вклучени во проектот, но не се девелопери, туку домен експерти, полесна манипулација со самите објекти преку користење на методи кои еднаш ги имплементираме и потоа ги реискористуваме.

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

- завршена работна задача (од task агрегатот кон project агрегатот се праќа настан за секоја завршена работна задача, откако ќе се прими настанот се проверува дали се завршени сите работни задачи на соодветниот проект за којшто се извршува работната задача којашто е завршена; ако се завршени сите работни задачи за проектот, се означува тој проект како завршен – се поставува вредноста на атрибутот на проектот finished на true)
- завршен проект (во project агрегатот се испраќа известување од ентитетот проект до вработениот кој управувал со проектот што е завршен)
- извршено плаќање на проектот (од payment агрегатот кон project агрегатот при секое плаќање се праќа настан за да се провери дали е платена целата сума за проектот (бидејќи плаќањето може да биде на рати); ако е платена целата сума, на проектот се поставува статус платен (payed=true) )
- промена на статус на проектот во finished (од project агрегатот кон client агрегатот - кога ќе се промени статусот на проектот во платен се праќа настан на клиент ентитетот за да се провери дали клиентот има сè уште некој неплатен проект; ако нема, неговиот статус isActive се поставува на false)