Projekti juhend

Harjutustundides 3–5 toimub iseseisva töö projekti tegemine. Kiire edenemine tunnis tähendab, et väljaspool tunde on vaja vähem teha ning suureneb tõenäosus saada projekti eest arvestus varakult ja sellega seoses saada ka lõpphindesse hea projekti kordaja. Ma soovitan Teil sellel perioodil teha projekti ka väljaspool harjutustunde, et kiiremini edasi jõuda ja rohkem oma vahetulemusi õppejõule ette näidata.

Projekti dokumendi jaotises 1.1 tuleb Teil fantaasiat ja taustateadmisi rakendades kirjeldada tervet infosüsteemi (seada paika selle piirid ning leida alamosad). Selleks tuleb Teil läbi mõelda, millega täpselt Teie organisatsioon tegeleb, kust tuleb kasum, millist lisandväärtust pakutakse selle süsteemiga suhtlejatele. Alates punktist 1.2 tegelete süsteemi ühe alamosaga.

Vormistamine

Eesmärk on, et Teie kirjatöö vormistus järgiks võimalikult täpselt lõputööde kirjutamise juhendit, mille leiate aadressilt: http://www.ttu.ee/teaduskond/infotehnoloogia-teaduskond/it-tudengile/loputoo-ja-lopetamine-9/loputoo-vormistamine-4/ Muuhulgas nõuab see dokument järgnevat.

- Kõik esimese taseme pealkirjad peavad algama uuelt leheküljelt.
- "Kõigis peatükkides ja alapeatükkides peab pealkirjale järgnema tekst. Uue alapealkirja, joonise või tabeli esitamine koheselt pärast pealkirja ei ole lubatud – pealkirjale järgneval real on alati tekst."
- Kõikidel joonistel on allkirjad ning tabelitel on pealkirjad.
- Kõikidele joonistele ja tabelitele viidatakse töö tekstis.
- Kõikidele kasutatud materjalidele viidatakse töö tekstis.
- Hoidutakse neljanda taseme pealkirjadest.

Projekti töövihik ja näiteprojekt järgivad neid nõudeid.

Töökorraldus harjutustundides

Projekti rühmatööna tegijad peavad tegema erinevaid tegevusi (analüüs, disain, programmeerimine, testimine), rühmas ei ole kindlaid rolle ning rühm kui tervik vastutab õigeks ajaks ja hea kvaliteediga tulemuse saavutamise eest. See on efektiivne ja tänapäeval levinud arendustöö tegemise viis (vt näiteks tuntud paindmetoodika Scrum arendajate rühma põhimõtteid: https://www.scrum-institute.org/Scrum_Roles_The_Scrum_Team.php).
Projekti tegemisel ei ole kohustust Scrumi põhimõtteid järgida, kuid see pole ka keelatud. Juhend ei kirjuta ette kasutatavat arendusmetoodikat.

Projekti tegijad võivad olla erinevatest rühmadest ja käia erinevates harjutustundides. Oluline on, et ühes harjutustunnis tehtu antaks järgmisesse tulijale edasi, et ta saaks seal pooleli jäänud kohast tööd jätkata.

Kui ühes tunnis käib mitu sama projekti tegijat, siis töö peaks meenutama teataval määral ekstreemprogrammeerimisest tuntud paaris-programmeerimise praktikat:

- https://en.wikipedia.org/wiki/Pair_programming
- https://www.infoq.com/articles/introducing-pair-programming

Üks kirjutab/loob diagramme/programmeerib. Teised jälgivad, juhendavad, parandavad, mõtlevad kaasa. Vahetage vähemalt kord 30 minuti jooksul kirjutamisega tegelejat, et ühe sessiooni jooksul saaksid kõik projekti liikmed vähemalt korra kirjutamisega tegeleda.

Ettevalmistumine

Seaduste mitte tundmine ei vabasta nende täitmisest. Infosüsteem peab olema seadustega kooskõlas, võimaldama sellest huvitatud osapooltel seaduseid järgida ning eriti hea, kui takistaks seaduste vastu eksimist.

- Teie andmebaasis on kindlasti isikuandmeid. Järelikult peab süsteem järgima isikuandmete kaitse seadusest tulenevaid nõudeid: https://www.riigiteataja.ee/akt/IKS. Muuhulgas nimetab seadus delikaatsed isikuandmed ning sätestab nõuded isikuandmete töötlemisele. Seaduse täitmise üle valvab Andmekaitse Inspektsioon, mille kodulehelt leiab hulgaliselt juhiseid, kuidas seadust õigesti järgida: http://www.aki.ee/et/juhised
- 2018. aasta mais jõustub Euroopa Liidu (EL) taseme isikuandmete kaitse üldmäärus. Määruse kohaselt on isikuandmeteks igasugune teave tuvastatud või tuvastatava füüsilise isiku kohta. Määrus rakendub kõigile EL liikmesriikidele otse, selleks ei muudeta siseriiklikke seaduseid.
  - http://arileht.delfi.ee/news/triniti/andmete-maaratlus-muutub-ehk-mida-peetakse-alates-2018-aastast-isikuandmeteks?id=75700843
  - http://arileht.delfi.ee/news/triniti/andmekaitsemaarus-muutub-tootlemiseks-vajaliku-andmete-omaniku-nousoleku-tingimused-karmistuvad?id=75871155
- Samuti heitke pilk peale tarbijakaitseseadusele, mis enamike Teie projektide valdkonda (kaubandus/teenindus – põhiobjektid *kaebus* ja *garantii)* puudutab: https://www.riigiteataja.ee/akt/TKS
- Interneti vahendusel tehtud ostude korral võib vastavalt võlaõigusseadusele lepingust taganeda 14 päeva jooksul ajast, mil toode kliendini jõudis. Teenuse ostu puhul algab tähtaeg lepingu sõlmimisest: https://www.tarbijakaitseamet.ee/et/tarbijale/14-paevane-taganemisoigus Seega on selliste süsteemide üks põhiobjekt *tagastus* või *loobumine.* USA kogemus näitab, et internetiostudest tagastatakse umbes kolmandik:
  - http://arileht.delfi.ee/news/uudised/netimuugi-ajastu-ootamatu-probleem-kolmandik-kaupu-saadetakse-tagasi?id=80055052

Samuti vaadake palun, kas Teie projekti valdkonnas on valmistarkvara, mis võiks projekti jaoks ideid anda.

- http://www.rmp.ee/tarkvara/
- https://www.odoo.com/
- https://blog.capterra.com/the-top-5-free-inventory-software-systems/

- http://www.softwareshortlist.com/website/software/

Tarkvara näitena vaadake palun e-poodi (nii kliendi kui tagakontori (ingl *back office*) osa): http://demo.prestashop.com/en/?view=front Kaupade/toodete haldus on selles tarkvaras alajaotuses *Catalog management*. Samuti võib vaadata Odoo tarkvara demo (https://demo3.odoo.com/web#home). Erinevad moodulid realiseerivad suure hulga infosüsteemi võimalikest funktsionaalsetest allsüsteemidest ja registritest.

Käesolev aineprojekt on üks viis infosüsteemi ja andmebaaside projekteerimise õppimiseks. Kuid kui seisate kunagi ülesande ees automatiseerida tööd ja võtta selleks kasutusele (uus) tarkvara, siis tuleb kõigepealt otsida turult sobivat tarkvara, mida kas otse kasutusele võtta või võtta kasutusele peale kohandamist. Kohandamine eeldaks kas kokkulepet tarkvara autoritega, kes seda teie eest teeksid, või avatud lähtekoodiga tarkvara kasutuselevõttu, koos omapoolsete paranduste tegemisega. Kui kohandamine tähendab puhtalt mingi lisamooduli loomist, mis loeb ja muudab andmeid ühises andmebaasis, siis pole eelnev isegi vajalik. Alles siis kui sellist tarkvara ei leita on mõtet hakata ise uut tarkvara tegema. Väidan, et enamike selle aine teemade jaoks on selline tarkvara olemas, seega päriselus sellise uue tarkvara tegemiseks peab olema väga hea põhjus (innovatsioon).

Need, kes teevad projekti **variandi 2 (vaba teema)** järgi, peavad minema http://maurus.ttu.ee/346 *Andmebaasid 1 => Iseseisva töö tegemise samm-sammuline juhend*. Sealt leiate dokumendi malli ning detailse juhendi. Juhendi samm 7 selgitab, kuidas sellist projekti nullist tegema hakata. See on kasulik info ka töövihiku kasutajaile, kes tahavad tulevikus näiteks tööga seotud andmebaasi ja andmebaasirakendusi sarnasel viisil projekteerida. **Töövihiku järgi projekti tegijad peavad jätkama selle dokumendi lugemist.**

Edasine kirjeldus on neile, kes teevad projekti variandi 1 (töövihik) alusel.

Töövihik on mõeldud tehingutöötlusele orienteeritud registri tüüpi infosüsteemi ühe põhiandmete haldamise funktsionaalse allsüsteemi ning selle vajatavate registrite projekteerimiseks.

Väike mõistete selgitus.

- *Tehingutöötlusele orienteeritud* – Süsteemi põhilised operatsioonid on üksikute olemite andmete lugemine ja muutmine. Sellist tüüpi süsteemi operatsioonide näiteks on üksiku tellimuse kinnitamine või üksiku arve lugemine.
- *Registri tüüpi infosüsteem* – Infosüsteem, mille põhiülesandeks on pakkuda võimalust andmeid talletada ja kasutada. Selle infosüsteemi tarkvara koosneb vormidest/lehtedest, mille kaudu hallatakse ühes või mitmes andmebaasis olevaid andmeid.
- *Põhiandmed* – Andmed mingite selliste abstraktsete või füüsiliste olemite kohta, millega seoses toimuvad süsteemis tehingud e transaktsioonid. Teiste sõnadega, need olemid loovad tehingutele konteksti. Näiteks selleks, et süsteemis saaks teha tellimusi (protsess), on süsteemil vaja andmeid klientide, kaupade ja tellimusi menetlevate töötajate kohta (põhiandmeid).

Jägnevalt nimetatakse kõige kriitilisemad sammud ja nende järjekord. Siin ei ole nimetatud kõiki dokumendi täiendusi. Kuna iga rühm töötab omas tempos, siis on need kõik ühte dokumenti kokku koondatud. **NB!** Kui X ei ole Teie töö lausetes sobivas käändes, siis järelikult olete lihtsalt teinud *Search and Replace*, dokumenti mitte süvenenud ja sellist tööd ei saa arvestada. X ainsuse ja mitmuse vormi valikul on oluline *järjekindlus* – sama tüüpi elemente (nt allsüsteemid, kasutusjuhud, olemitüübid, tabelid) tuleb nimetada ühes stiilis. Öeldakse, et arvutiteaduses on ainult kaks keerulist teemat, millest üheks on nimede andmine: https://martinfowler.com/bliki/TwoHardThings.html

Töövihikus sisaldub:

- fikseeritud osa,
- muudetav osa, mille tunneb ära:
  - X
  - ...
  - pealkiri, mille alla pole midagi kirjutatud

Töövihikus ei ole lubatud fikseeritud osi kustutada või muuta (v.a juhul kui muutmise nõue või võimalus on kirjas tööjuhendis). Töövihik on kui rakendusraamistik (*application framework*) (https://en.wikipedia.org/wiki/Application_framework), milles arendaja peab "liha luudele" lisama. See kirjeldab erinevate projektide ühiseid osi ning jätab arendaja ülesandeks kirjeldada, mis neid eristab. Töövihiku eesmärk on saada kiiresti valmis süsteemi funktsionaalne kontekst, et saaks keskenduda andmebaasi kirjeldamisele ja realiseerimisele. **Kõigile töövihiku alusel projekti tegijatele on järgneva juhendi järgimine kohustuslik.** Mida, mis järjekorras ja kuidas teha räägitakse ka loengutes, mille lindistuste viited leiate kataloogist *Teooria testideks (vahetestid, eksam) valmistumine*.

Mida ja mis järjekorras töövihiku kasutamise korral teha?

1. Töövihiku esmane ülevaatamine ja täiendamine

Alustage palun projekti tegemist sellest, et:

- võtate ette malli töövihiku dokumendi ning mudelite põhju sisaldava CASE faili,
- vaatate/loete need algusest lõpuni järjest süstemaatiliselt läbi ja üritate aru saada,
- asendate **IGAL POOL** X sobivas käändes põhiobjekti nimega (X-i valiste ära projekti teema registreerimisel).
  - Ainsuse/mitmuse vormi valimisel (nt allsüsteemide nimetamisel) vaadake palun töövihikus juba kirjas olevaid näiteid ja kasutage sama stiili.
  - Dokumendifailis muutke palun ka tiitellehte.

Peale seda, kui olete asendanud X-d CASE failis, saate kohe dokumenti ära kopeerida järgmised diagrammid:

- tegevusdiagramm (paketis *Analüüs/Funktsionaalsed allsüsteemid*),
- registri kontseptuaalne eskiismudel (paketis *Analüüs/Registrid*).

Kopeerimiseks piisab kiirkorraldustest Ctrl + A (vali kogu diagramm); Ctrl + C (kopeeri); Ctrl + V (kleebi). Kui see lähenemine mõnes keskkonnas (nt Google Docs) ei tööta, siis tuleks diagramm kas pildina eksportida (https://stackoverflow.com/questions/10917228/enterprise-architect-export-uml-diagrams-in-high-quality) või teha diagrammist ekraanitõmmis (kasutan ise tarkvara https://mwsnap.en.softonic.com/, kuid kindlasti on ka teisi).

Ülejäänud diagramme (kasutusjuhtude diagramm, olemi-suhte diagrammid, seisundidiagramm) tuleb siiski rohkem täiendada (vt järgmised punktid) ja nende kopeerimisega tuleb oodata. CASE projekti süstemaatiline täiendamine tähendab, et võtate ülevalt-alla lahti kõik paketid (kaustad) ja alampaketid ning vaatate üle ja muudate kõiki sealt leitud diagramme.

Kui teete projekti mitmeksesi, siis protsessi kiirendamiseks saate töötada samaaegselt – üks täiendab dokumenti, teine diagramme ja kui on kolmas, siis tema jälgib ja juhendab üle õla kummagi tööd. Kuna diagrammidega läheb ilmselt kiiremini, siis kui need on täiendatud, jätkate dokumendiga paarisprogrammeerimise stiilis. Teine võimalus tootlikuse parandamiseks on, et rühma liikmed käivad 3–5 nädalal erinevates harjutustundides ja annavad oma tunnitöö tulemuse järgmisele tegijale üle.

Hiljem peaksid kõik projekti tegijad dokumendi ja CASE projekti üksinda ülevaatama, et:

- leida ja asendada asendamata jäänud X-e,
- saada ise aru, millised mudeli osad kus asuvad.

Oranžil taustal olev tekst puudutab dokumendi osa, kus kirjeldate **tervet süsteemi**.

2. Täiendage organisatsiooni eesmärke.

Kui vaadeldav organisatsioon *ei taotle kasumit* (nt tegeleb heategevusega), siis tuleb eesmärkide hulgast kustutada kasumi taotlemisele viitav eesmärk.

Juhendi punktide 3–7 korral on töövihikusse kirja pandud infot süsteemi administratiivsete alamosade kohta, ilma milleta süsteem ei saa toimida ja mis on ühised paljude erinevate valdkondade süsteemidele. Mõned sellised osad võivad siiski olla töövihiku eeltäidetud osast puudu. Põhiobjekte (v.a *Isik, Töötaja, Klassifikaator, Klient, X*) võib ka töövihikust kustutada, kuid see eeldab nendele põhiobjektidele vastavate sissekannete kustutamist kõigist töövihiku dokumendi jaotise 1.1 osadest (vt järgneva juhendi punkte 3–7).

Üldiselt aga tuleb keskenduda süsteemi sisuliste alamosad leidmisele, mis annavad sellele süsteemile "oma näo".

3. Põhiobjektide nimekirja täiendamine. Kasutage põhiobjektide leidmiseks *integratsiooni mustri* abi. Toon järgnevalt näiteid võimalikest puuduvatest põhiobjektidest. Mõni järgnevalt mainitud põhiobjektidest on siiski juba töövihikus kirjas – ärge neid teist korda kirja pange.

Selleks, et kavandada *tarkvara*, peavad meil olema *nõuded tarkvarale*. Selleks, et esitada *nõudeid*, on vaja tunda *valdkonda*, milles see tarkvara tööle hakkab. (Dines Bjørner)

- Kui ettevõttes toimub graafikujärgne töö, siis on põhiobjektide hulgas:
  - *Töögraafik*
  - *Töötamine*
- Organisatsioonile võivad teha ettekirjutusi erinevad riiklikud ja kohaliku omavalitsuse organisatsioonid. Seega on üks põhiobjekt:
  - *Ettekirjutus.*
  - Toon näiteid.
    - Toitlustusasutustele võib teha ettekirjutusi Terviseamet (http://www.terviseamet.ee/info/oluline-info/ettekirjutused-ja-trahviotsused.html).
    - Kui organisatsioon töötleb isikuandmeid, siis nende töötlemise nõuete vastu eksimisel teeb ettekirjutusi Andmekaitse Inspektsioon (http://www.aki.ee/).
- Kui organisatsioon (nt kauplus, teenindusasutus, toitlustusasutus, majutusasutus) esitab avalikult muusikapalasid (isegi raadiost või TV-st), siis tuleb selle eest maksta autoritasu (http://www.eau.org/see-on-eau/muusikateoste-autorid-ja-kirjastajad/teoste-avalik-esitamine/). Eesti Autorite Ühingu kodulehel tuleb täita repertuaari aruandlust (http://www.eau.org/autorile/muusika-autor/repertuaari-aruandlus/). See tähendab, et põhiobjektide hulgas peab olema:
  - *Muusikapala esitus*
  - Võibolla ka *Muusikapala* ja *Autor*
  - *Leping* (organisatsioon sõlmib "ühe Eesti Autorite Ühinguga ja teise (olenevalt valdkonnast, kus muusikat kasutatakse) kas Eesti Esitajate Liidu või Eesti Fonogrammitootjate Ühinguga.

Eesti Esitajate Liit ja Eesti Fonogrammitootjate Ühing teostavad oma õiguseid selles valdkonnas ühiselt." (http://www.efy.ee/index.php?page=118)).

- Eriti suuremates organisatsioonides võidakse teha auditeerimisi. Auditi eesmärk on teha kindlaks uuritava objekti (nt organisatsiooni või mõne selle pakutava toote/teenuse) vastavus nõutavatele kriteeriumidele. https://et.wikipedia.org/wiki/Audit Auditeerimisega peaks tegelema mõni sõltumatu osapool ja see teenus ostetakse organisatsiooni audiitorfirmalt sisse. Põhiobjektide hulka tuleks lisada:
  - *Audit*
- Tark õpib teiste vigadest. Võibolla oleks kasulik koguda infot tüüpprobleemide ja nende lahenduste kohta (nagu Mauruse keskkonna *Helpdesk* osa). Sinna alla kuuluvad nii töötajate kui klientide probleemid. Sellisel juhul on põhiobjektide hulgas:
  - *Tüüpprobleem*
- Kui organisatsioonis on vaja vahendada klientidele või töötajatele infot oluliste sündmuste kohta, siis on põhiobjektide hulgas:
  - *Sündmus (võib ka kasutada nime Uudis)*
- Infosüsteemi piiride määramiseks ja põhiobjektide leidmiseks tuleb läbi mõelda, milliseid tegevusi teevad organisatsiooni töötajad ise ning milliseid teenuseid ostetakse organisatsiooni allhankena (https://et.wikipedia.org/wiki/Allhange) või tugiteenusena sisse. Protsesside ning füüsiliste/abtraktsete asjade üle, millega organisatsiooni töötajad ise tegelevad, peab organisatsiooni infosüsteem kindlasti arvet pidama ning seega tuleb leida neile vastavad põhiobjektid. Protsesside ning füüsiliste/abtraktsete asjade korral, millega tegelevad teenusepakkujad või allhankijad, tuleb mõelda, kas info nendest peab jõudma ka projekteeritavasse infosüsteemi. Kui jah, tuleb leida vastavad põhiobjektid, kui ei, siis mitte.
  - Üheks parkla infosüsteemi põhiobjektiks võib olla *Teisaldamine*: http://www.delfi.ee/news/paevauudised/krimi/riigikohus-otsustas-et-eraparklal-on-oigus-parkimistasu-maksmata-jatnud-omaniku-auto-teisaldada?id=74040871 Kui parkla võib sõiduki teisaldada, siis küsimus on, kes seda teeb – kas töötaja või ostab parkla teenust sisse. Kui ostab teenust sisse, siis on põhiobjektid ka *Partner* ja *Leping* (partneritega tuleb sõlmida lepingud). Kui sõidukit ei teisaldata, siis hoiab see teenuse osutamise vahendit (parklakohta) kinni ja pole kindel, kas raha hiljem võlglaselt kätte saab. Isegi kui teisaldamist teeb teenusepakkuja peab parkla infosüsteemi selle kohta jälg jääma (seega on põhiobjekt *Teisaldamine*), et saaks kliendile öelda, kuhu tema sõiduk kadunud on. Igal juhul tuleb parkimistingimuste rikkujalt sisse nõuda trahv ja põhiobjekt on ka *Trahv.*
  - E-poed võivad lao ning kohaletoimetamise teenuse sisse osta või sellega ise tegeleda. Siin on hea näidetega eestikeelne ülevaade mõlema lahenduse eelistest ja puudustest ning realiseerimisest:
    - https://www.aripaev.ee/uudised/2010/11/29/e-pood-see-on-imelihtne

- Näiteks e-pood võib ajada äri nii, et vajab vaheladu. Seda läheb vaja, et tarnijatelt tulnud saadetisi hoiustada, ümberpakkida ja kliendile edasi saata. Põhiobjektide hulgas on näiteks:
  - *Kauba laoliikumine*
  - *Inventuur*
  - *Saadetis*
- Teisalt võib see ajada äri nii, et tellimused saadetakse edasi tarnijale, kes siis paneb kokku saadetised ning need otse klientidele edastab. Sellisel juhul eelnevalt nimetatud põhiobjektid seda süsteemi ei iseloomusta. Sellise ärimudeli korral tekib küsimus, miks peaks keegi üldse sellist vahendajat vajama? Võibolla on ärimudel selline, et e-poe pidaja pakub erinevatele tarnijatele e-poe teenust, st tarnijad ei pea selle püstipanemise ja ülalhoiuga ise tegelema. Sellisel juhul on ka e-poe pidaja infosüsteemi üks põhiobjekt:
  - *Teenus* (teenuse näide – E-pood kuni 1000-le kaubale ja kuni 20000 kliendile kuus maksab X EUR kuus).

- Restoranis on vaja pidada arvestust laudade üle (põhiobjekt *Laud),* et tellimus elektrooniliselt vastu võtta, lauaga siduda ning ettekandjatel oleks näiteks elektroonilisel tablool näha, millisesse lauda tellimus kohale viia: http://kasulik.delfi.ee/news/uudised/uuenduslik-vaikelinna-restoran-volub-kliente-ainulaadse-teenindamisega?id=77135412 Sama põhimõtet saaks järgida parklas. Parkimiskohal on andurid, mis annavad infot, kas koht on vaba või mitte. Parkla sissepääsu juures on tabloo, mis näitab vabade kohtade asukohta.
- Tootmisettevõtte puhul on põhiobjektide hulgas näiteks:
  - *Hinnapäring*
  - *Hinnapakkumine*
  - *Tootmistellimus*
  - *Tootmine*
  - *Materjal*
  - *Materjali tarnetellimus*
  - *Materjali laoliikumine*
  - *Reklamatsioon* – sisuliselt kliendi kaebus, et toode ei vasta standarditele või tehnilistele tingimustele: http://www.lkdiislikeskus.ee/tmmaailm2.htm#Reklamatsioon
  - *Kvaliteedikontroll* – Vaja on kontrollida nii sisendiks olevaid materjale ning pooltooteid, ettevõttes toimuvaid protsesse kui ka tootmise tulemust. See on pidevalt toimuv protsess. http://www.lkdiislikeskus.ee/tmmaailm2.htm#Kvaliteedikontroll
  - *Rike* – tootmiseks kasutatavate seadmete mittevastavus nõuetele. http://www.lkdiislikeskus.ee/tmmaailm2.htm#Rike
  - *Tööõnnetus*
  - *Praak*
- Mõelge sellele, kust saab organisatsioon tulu.
  - Kui taotletakse kasumit, siis on näited võimalikest põhiobjektidest:
    - *Kliendi tellimus*
    - *Reklaami tellimus* (partnerid saavad tellida reklaame, mida kuvatakse ettevõtte veebilehel või ettevõtte ruumides),

- *Reklaam*
- Kui kasumit ei taotleta, kuid ettevõtmist finantseeritakse annetustest, siis on põhiobjekt näiteks:
  - *Annetus*

- Integratsiooni mustri kohaselt on X kas *ressurss/vara* (väljak, rada, parklakoht, laud, peatus, kabiin, hoiuruum, kaart, sõiduk, raamat, video), *toode* (teenus, kaup, toode, treening), *asukoht* (väljak, rada, parklakoht, laud, peatus, kabiin, hoiuruum) või *sündmus* (üritus, kampaania). Nagu näte võib põhiobjekt kuuluda rohkem kui ühte kategooriasse. Mõelge põhiobjektide leidmiseks nendega seotud sündmustele ja ülesannetele.
  - Näiteks parklakohaga on seotud sündmused nagu *parkimine, teisaldamine* ja *trahvimine.*
- *Juuksurisalong, spordiklubi, kauplus, restoran,* ... on põhiobjektid **vaid siis**, kui tegemist on nende ketiga. Ketis võib olla mitu lüli ja iga lüli kohta on vaja andmeid koguda.
- Leidke analoogiaid. Toon näiteid.
  - Spordiklubis treenitakse keha, ülikoolis vaimu. Enamikele ülikooli infosüsteemi põhiobjektidele (õppekava, õpingukava, aine, tunniplaan, rühm, ruum jne) võib olla analoog ka spordiklubi infosüsteemis.
  - Kauba laoseisu analoog on kontojääk pangas. Kontojääk leitakse kontoga seotud tehingute põhjal, kauba laoseis leitakse kaubaga seotud laoliikumiste e laokannete e laotehingute põhjal. Pangakonto analoogiks on kaup. Seega e-poe/poe/hulgilao infosüsteemis on põhiobjektide hulgas:
    - *Kaup*
    - *Kauba laoliikumine*

4. Tegutsejate nimekirja täiendamine.
   - Nimetage vaid otseselt infosüsteemi tarkvara kasutavad tegutsejad. Mida detailsem tegutsejate nimekiri, seda parem. Tegutseja = roll – mida rohkem on rolle, seda suhteliselt vähem on iga rolliga seotud ülesandeid. Mida "väiksemad" on rollid, seda paindlikum (parem) on õiguste jagamine.
     - Mõelge iga põhiobjekti korral, milline tegutseja hakkab sellele vastavaid andmeid haldama. Vajadusel lisage tegutsejaid (rolle). Kui näete, et mingile tegutsejale tekib niimoodi liiga palju ülesandeid, siis see on see selge märk, et tegutsejaid tuleb juurde lisada.
     - Rollid ei pruugi langeda kokku hetkel kasutusel olevate ametikohtadega, kuid ametikohtade nimekiri on siiski piisavalt hea lähendus esialgse rollide nimekirja koostamiseks.
       - https://www.thebalance.com/careers-by-field-or-industry-526085
         - Restoran: https://www.thebalance.com/restaurant-job-titles-2061543
       - http://ametid.rajaleidja.ee/
     - "Töötaja" on tegutsejate nimekirjas vaid siis, kui on mingi hulk süsteemi pakutavaid funktsionaalsuseid, mis on ühised kõigile töötajatele – alates esimest päeva tööl olev assistendist kuni tippjuhini välja.
     - Kui tegemist on kaupu või teenuseid pakkuva organisatsiooniga, siis kas maksete tegemiseks kasutatakse pangalinki?

    - Kui jah, siis on *Pank* üks tegutsejatest, sest peale Teie infosüsteemi algatatud makse teostamist saadetakse panga poolt Teie infosüsteemi koheselt vastussõnum, millega pank kinnitab, et klient on tellimuse eest tasunud.
    - Kui tasumine aga käib nii, et klient teeb makse pangas ning organisatsiooni sobivas rollis töötaja jälgib internetipangas laekunud makseid ja märgib tellimuse makstuks, siis *Pank* ei ole tegutseja (sest ei suhtle otse Teie infosüsteemiga).
5. Põhiobjektide nimekirja alusel funktsionaalsete allsüsteemide ja registrite leidmine.
    - Põhiobjektide ja funktsionaalsete allsüsteemide vahel on üks-ühele vastavus.
    - Põhiobjektide ja registrite vahel on üks-ühele vastavus.
    - Kuidas eristada sisulisi ja administratiivseid funktsionaalseid allsüsteeme/registreid
        - https://martinfowler.com/bliki/UtilityVsStrategicDichotomy.html
        - Sisuline allsüsteem vastab *strateegilisele tarkvarale* (*strategic software*) ja administratiivne allsüsteem tarbetarkvarale *(utility software)*.
    - Pöörake tähelepanu, et allsüsteemide nimedes kasutataks läbivalt mitmuse vormi.
        - **Halb**: Laua funktsionaalne allsüsteem/register
        - **Parem**: Laudade funktsionaalne allsüsteem/register
    - Pöörake palun tähelepanu, et allsüsteemile viitamiseks kasutatakse kogu projektis ühesugust nime:
        - **Halb**: Dokumendis ja CASE projektis on segamini nimed nagu: parkimiskoha register, parkimiskohtade register, parklakoha register, parklakohtade register
6. Tegutsejate nimekirja alusel pädevusalade leidmine.
    - Tegutsejate ja pädevusalade vahel on üks-ühele vastavus.
7. Täita ülejäänud punkt 1.1, mille saate suures osas täita tegutsejate ja põhiobjektide nimekirja alusel.
    - *Iga põhiobjekti kohta* **vähemalt üks** *infosüsteemi eesmärk*.
    - *Iga põhiobjekti kohta* **vähemalt üks** *lausend,* kus seda nimetatakse.
    - *Iga põhiobjekti kohta* **üks kuni kolm** *põhiprotsessi,* mis Teile esimesena sellega seoses pähe tulevad.
    - *Iga kirja pandud põhiprotsessi kohta* **üks või rohkem** *sündmust* (põhjust), mis on protsessi käivitamise päästikuks.
    - Jälgige, et *iga tegutsejat* nimetataks samuti **vähemalt ühes** *lausendis*.
    - Asukohtade juures mõelge, kas mõni tegutseja peaks mõne põhiprotsessiga seoses kasutama mobiilset seadet (nt nutitelefoni või tahvelarvutit). Samuti mõelge, kas klientidega suhtlemiseks võiksid olla kasutusel iseteeninduse kioskid.
        - Näiteks restoranis saaksid kelnerid mobiilse seadme abil registreerida tellimusi ja arvete maksmisi. Näiteks parkla valvurid saaksid mobiilse seadme abil registreerida parkimisreeglite rikkumisi ja teisaldamisi.
        - Näiteks treeningkeskuses saaksid kliendid osta pileteid iseteeninduse kioskist.
        - Mõelge asukohtadega seoses ka sellele, kas süsteem peaks võimaldama nende asukohtade või kasutatavate tehniliste

vahendite üle arvestust pidada. Kui jah, siis vaadake palun, et põhiobjektide nimekirjas oleks olemas selle jaoks sobivad põhiobjektid (nt *ruum*, *vara).*

8. Seisundidiagrammile sündmuste lisamine. Seisundidiagramm on paketis *Analüüs/Registrid*.
   - Kuna igale seisundi üleminekule vastab üldjuhul eraldi kasutusjuht, siis leiate info seisundiüleminekuid käivitavate sündmuste kohta neile vastavate kasutusjuhtude laiendatud formaadis tekstikirjeldustest. Saate need sealt seisundidiagrammile kopeerida.
   - Pöörake palun sündmuste kirjeldamisel tähelepanu, et kasutusjuhtude mudeli kohaselt tegeleb X lõpetamisega juhataja.
9. Kasutusjuhtude diagrammi võimalik täiendamine (valikuline). Kasutusjuhtude diagramm on paketis *Analüüs/Funktsionaalsed allsüsteemid*. Peale seda võite diagrammi dokumenti kopeerida.
   - Kui leiate, et kasutusjuhtude mudelis peaks mõne olemasoleva kasutusjuhuga (peale *Tuvasta kasutaja*) olema seotud veel mõni Teie kirja pandud tegutseja, siis siduge see tegutseja soovitud kasutusjuhtudega nii kasutusjuhtude diagrammis kui ka kasutusjuhu kõrgtaseme ja laiendatud formaadis kirjelduses. Ärge unustage kirjeldada kasutusjuhu laiendatud formaadis selle tegutseja huvisid antud kasutusjuhuga seoses. Kui sellel tegutsejal on vaja selle kasutusjuhu läbimiseks ennast süsteemil tuvastada lasta, siis siduge see tegutseja ka kasutusjuhuga *Tuvasta kasutaja*. Lisaks tuleb seda tegutsejat nimetada punktis *1.2.2 Allsüsteemi kasutavad pädevusalad.* CASE projektis tuleb tegutseja panna paketti *Pädevusalad*. Samuti tuleb siis see tegutseja lisada CASE projekti diagrammile *Äriarhitektuuri fragment (detailse projekti skoop).*
     - Näiteks treeningute andmeid peaks saama vaadata treener, radade andmeid instruktor, pakutavate teenuste andmeid teenindaja.
10. Kontseptuaalse andmemudeli täiendamine.
    - Tulenevalt kasutusjuhtude mudelist on vaja lisada olemitüüp *Klient*, sest klient peab saama süsteemi sisse logida. Kliendi all mõeldakse eraklienti. Kliendi üldistus on *Isik. Klient* ja *Töötaja* on *Isiku* rollid, st modelleerige *Klient* samamoodi nagu on töövihikus on modelleeritud *Töötaja.* Näidake, et vastavad üldistusseosed moodustavad ühise üldistuste hulga (*generalization set*) ning täpsustage selle kitsendused. Näidake neid kitsendusi diagrammil.
    - Lisage registrivaatesse pakett *Klientide register* ja paigutage olemitüüpi *Klient* esitav klass sellesse paketti.
      - Leidke CASE projektist diagramm *Äriarhitektuuri fragment (detailse projekti skoop)* ning väljendage sellel, et *X funktsionaalne allsüsteem* vajab tööks *Klientide registrit*.
    - Tulenevalt kasutusjuhust *Tuvasta kasutaja* peab iga kliendi korral saama registreerida tema hetkeseisundi. Kui klient on ebasobivas seisundis, siis ta sisse logida ei saa. Kasutage selle olemi-suhte diagrammil kujutamiseks sama mustrit, mida rakendatakse isikute, töötajate ja X-de hetkeseisundi registreerimiseks.
      - Olemitüüp *Kliendi_seisundi_liik* tuleb paigutada paketti *Klassifikaatorite register* ning see tuleb esitada *Klassifikaatorite registri* olemi-suhte diagrammil. Üldistusseose modelleerimisel

ärge unustage lisada seda olemitüübiga *Klassifikaator* seotud üldistuste hulka (*generalization set*).

- Tulenevalt ärireeglitest peab iga kliendi korral saama registreerida, kas ta on nõus oma tarbijaharjumuste uurimisega või otseturundusega või mitte (kliendi atribuut *on_nous_tylitamisega : Boolean*). Vaikimisi andmekaitse põhimõttest lähtuvalt tuleb vaikimisi eeldada, et klient ei ole sellega nõus.
  - Isikuandmete kaitse seadus:
    - https://www.riigiteataja.ee/akt/130122010011 § 12, lõige 5.
  - Vaikimisi andmekaitse põhimõte 2.1.1 Euroopa Liidu isikuandmete kaitse üldmääruses
    - http://www.aki.ee/et/andmekaitse-reform/vaikimisi-ja-loimitud-andmekaitse
- Tulenevalt töö spetsiifikast tuleb määrata, milliseid andmeid töö teemaks oleva põhiobjekti e põhiolemitüübi kohta veel registreeritakse ning need kontseptuaalses andmemudelis korrektselt kirjeldada. **Ärge unustage kitsendusi – nende näited on töövihikus looksulgudes {}!**
  - Kohustuslik on lisada X-i vähemalt üks atribuut, mille väärtus esitab mingit X arvulist omadust.
  - Kohustuslik on lisada X-i vähemalt üks atribuut, mille väärtus esitab mingit X tekstilist omadust. *Näiteks* võib see olla nimetus, kirjeldus või kommentaar.
    - Kui lisate nimetuse, siis tuleb otsustada, kas selle väärtus peab olema iga X-i korral unikaalne või mitte.
  - Kohustuslik on on modelleerida vähemalt üks täiendav klassifikaator, mille väärtuseid kasutatakse X-ide iseloomustamiseks.
    - Klassifikaatorite registrisse kuuluvad olemitüübid tuleb paigutada paketti *Klassifikaatorite register* ning need tuleb esitada klassifikaatorite registri olemi-suhte diagrammil. Üldistusseoste modelleerimisel ärge unustage lisada neid olemitüübiga *Klassifikaator* seotud üldistuste hulka (*generalization set*).
      - Mis on klassifikaator? https://www.stat.ee/klassifikaatorid
  - Kõigi kolme eelneva korral on vaja otsustada, kas selle väärtuse registreerimine on iga X-i korral kohustuslik või mitte.
  - Kui teemaks on *parklakohtade arvestus* ning olete määranud, et üks põhiobjekt on *Parkla* (organisatsioon haldab mitut parklat), siis tuleb siduda parkla ja parklakoht.
  - Kui teemaks on restorani *laudade arvestus* ning olete määranud, et üks põhiobjekt on *Ruum* (restoranis võivad lauad olla erinevates ruumides), siis tuleb siduda ruum ja laud.
  - Teie projektis tähendab *Treening* treeningu spetsifikatsiooni ja *Kaup* kauba spetsifikatsioone e kaubaartikleid. See töövihik pole mõeldud treeningtundide või kauba eksemplaride arvestamise süsteemi loomiseks.
  - Kui teemaks on *treeningute arvestust*, siis vaadake ÕISist õppeaine IDU0220 (Andmebaasid I) ainekaardi kirjeldust. Treeningklubis treenitakse keha, ülikoolis vaimu. Treeningu (mis antud juhul tähendab treeningu spetsifikatsiooni) analoogiks ülikoolis on ainekaart (aine spetsifikatsioon). Vaadake ainekaarti

ja mõelge, kas selliseid andmeid oleks vaja hoida ka treeningu korral.

- Kui teemaks on *kaupade arvestus*, siis on sõltuvalt müüdavast kaubast võibolla lisaks müügihinnale vaja registreerida ning tarbijatele teatavaks teha ühikuhind. Seda nõuab tarbijakaitseseaduse paragrahv 7 ja nõudeid täpsustab majandus- ja taristuministri määrus "Kauba ja teenuse hinna avaldamise nõuded":
  - https://www.riigiteataja.ee/akt/111022016017
- Kui teemaks on *kaupade arvestus*, siis tuleb otsustada, mida kujutab endast kauba kood ning kas koode on rohkem kui üks. Mõelge sellele, et kui kauplus esitab tarnijale kaupade tellimuse, siis peab tarnija aru saama, milliseid kaupu talt tellitakse. Seda aitab saavutada selliste kauba koodide kasutamine, millest nii kaupmees kui tarnija ühtemoodi aru saavad.
  - GTIN (Global Trade Item Number) – globaalne kaubaartikli number (https://en.wikipedia.org/wiki/Global_Trade_Item_Number). GTIN koodist genereeritakse spetsiaalse tarkvara abil vöötkood. Kassas või laos loetakse vöötkoodi lugeja abil vöötkood ning sellesse kodeeritud info alusel leitakse andmebaasist kauba andmed (nt hind). Kui kaupa ei müüda hulgimüüjatele edasi ning vöötkoodi läheb vaja vaid sisemiseks kasutamiseks, siis ei pea kaubale GTIN koodi taotlema (http://www.barcodehq.com/upcnumber.html). Kuidas Eestis kaubale GTIN koodi taotleda? http://www.gs1.ee/?d=GS1_taotlemine
  - Amazoni lehel on hea ülevaade – Locating Product Identifiers (https://www.amazon.com/gp/help/customer/display.html?nodeId=200202190), kus räägitakse erinevatest kaupade identifikaatoritest. Amazon annab igale kaubale oma identifikaatori – Amazon Standard Identification Number. Kuid GTINe, mis on universaalsed standardiseeritud identifikaatorid, on vaja, et oleks võimalik vahetada infot teiste süsteemide ja organisatsioonidega. Seega, andmebaasis võib ühe kauba kohta olla ka mitu erinevat tüüpi koodi (kontseptuaalses andmemudelis vastaksid sellele erinevad atribuudid).
  - Veel üks kokkuvõtlik ülevaade toodete erinevatest globaalsetest identifikaatoritest: https://support.google.com/merchants/answer/160161?hl=en
- Vaadake lehel *schema.org* olevate andmekoosseisude kirjeldusi. Sealt võib saada inspiratsiooni, milliseid andmeid enda andmebaasi koguda. Palun olge tähelepanelikud – paljud seal põhimõistete juures kirjeldatud andmed hakkaksid kuuluma tegelikult teistesse registrisse ja jäävad Teie projekti skoobist välja. Paljudest seal toodud mõistetest ei teki mitte atribuudid, vaid eraldi olemitüübid ("kastid") ja seosed ("jooned").
  - Toode, Kaup, Kaubaartikkel: https://schema.org/Product
  - Teenus: https://schema.org/Service (ka treeningut saab vaadata kui teenust)

        - Golfirada: https://schema.org/GolfCourse
        - Auto: https://schema.org/Car
        - Teos: https://schema.org/CreativeWork
        - ...
    - Kui kasutate Enterprise Architecti, siis märkige olemi-suhte diagrammis, et järgnevate atribuudid on fakultatiivsed üheväärtuselised atribuudid (võimsustikuga 0..1, mitte 1 nagu on töövihikus kirjas):
        - Isik.eesnimi
        - Isik.perenimi
        - Isik.elukoht
    - Muutke olemitüübi *Isik* attribuutide järjekorda, liigutades atribuudi *e_meil* teiseks atribuudiks. Tehke sama ka dokumendis atribuutide definitsioonis. Põhjendus on, et (e_meil) on üks isiku unikaalne identifikaator ning selliseid atribuute on tavaks esitada esimeste seas.
    - Parema loetavuse huvides esitage olemitüüpide sõnaliste kirjelduste tabelis ja atribuutide sõnaliste kirjelduste tabelis read olemitüüpide nimede järgi tähestiku järjekorras sorteerituna.
    - Parema loetavuse huvides esitage atribuutide sõnalise kirjelduse tabelis olemitüübi atribuudid samas järjekorras nagu need on olemi-suhte diagrammil.
    - Asendage üks olemi-suhte diagramm mitme diagrammiga nii, et iga X funktsionaalse allsüsteemi tööks vajaliku registri kohta on projektis üks või rohkem eraldi diagrammi.
        - Vaadake palun näiteprojektidest, kuidas tulemus peab välja nägema.
        - Tükeldamise käigus ei tohi kaotsi minna algsele X registri diagrammile kirja pandud kitsendused. Uuel diagrammil tuleb need uuesti luua.
        - Klassifikaatorite registris võib olla nii paljude olemitüüpide andmeid, et parem oleks jagada need mitme diagrammi vahel.
    - Pange palun tähele, et etteantud kontseptuaalses andmemudelis tähendab *X_kategooria* **üksteist mittevälistavaid** kategooriaid. Näiteks oletame, et kirjeldate laua kategooriatena "ümmargune" ja "neljakandiline". Andmebaasi struktuur võimaldaks märkida sama laua nii ümmarguseks kui ka neljakandiliseks või siis vastupidi jätta laud sidumata ühegi kuju kirjeldava kategooriaga. Kui soovite laua puhul registreerida täpselt ühe kuju, siis on vaja defineerida *Laua_kuju* kui *Klassifikaator* ja siduda see üks-mitu seose kaudu lauaga (http://viktor.ld.ttu.ee/animatsioonid/animation_uml/).
        - Kategooriate kohta leidub standardeid.
            - Raamatud: http://www.bic.org.uk/7/bic-standard-subject-categories/
            - Tooted: http://www.gs1.org/gpc
11. Andmebaasioperatsioonide lepingute kooskõlastamine kontseptuaalse andmemudeliga.
    - Lepingutes jätke o väiketähed muutmata.
    - Pange tähele, et OP 4 on üldse lahti kirjutamata.
    - Pange tähele, et OP6 abil ei muudeta seisundit, registreerijat ega registreerimise aega! Seisundi muutmiseks on teised operatsioonid.

Registreerimise aeg ja registreerija peavad jääma paika, sest süsteem ei tohi võimaldada andmeid võltsida.

- Pange tähele, et OP6-le antakse ette vana ja uus kood, sest operatsiooniga võidakse ka muuta koodi. Vana koodi järgi leitakse üles muudetava olemi andmed ja selle atribuudi X_kood väärtuseks saab uus kood. Teiste atribuutide puhul pole vaja vana ning uut versiooni ette anda.
- Kui modelleerisite kontseptuaalses andmemudelis olemitüübi Y, mis kuulub samuti X registrisse ja mis on X-ga seotud nii:
  - `[X]-1--------0..*-[Y]`
  - `[X]-1--------0..1-[Y]`
  - `[X]-0..1--------0..*-[Y]`
  - `[X]-0..1--------0..1-[Y]`
  - , siis tuleb projekti mudelite osa täiendada. Järgnev kehtib ka X registrisse kuuluvate olemitüüpide korral mis on omakorda seotud Y-ga jne. Iga Y kohta tuleb lisada järgnevat. Selle tegemisel võtke eeskuju andmebaasioperatsioonidest OP7 ja OP8 ning nende kajastamistest teistes mudelites.
    - Andmebaasioperatsioonid Y andmete lisamiseks, kustutamiseks ja võibolla ka muutmiseks.
    - Viited nendele andmebaasioperatsioonidele tuleb lisada
      - seisundidiagrammi seisundi üleminekutesse – sinna kus kirjeldatakse reaktsioone sündmustele. Y andmete lisamine/muutmine/kustutamine on osa X andmete muutmisest.
      - kasutusjuhtude laiendatud formaadis tekstikirjeldustesse.
    - Nende andmebaasioperatsioonide tehtavad andmemuudatused peavad kajastuma CRUD maatriksis.

12. Seisundidiagrammil seisundi üleminekutes õigetele operatsioonidele viitamine. Need operatsioonid kirjeldavad, millised andmed, millise sündmuse tulemusena süsteemis registreeritakse. Peale seda võite diagrammi dokumenti kopeerida.
13. Kasutusjuhtude laiendatud formaadis tekstikirjeldustes ... asendamine, viidates andmetele, mille kirjelduse olete ise kontseptuaalsesse andmemudelisse juurde lisanud. Kui leiate, et kasutusjuhus juba kirjeldatud süsteemi poolt esitatavate andmete hulk on piisav, siis peab ... lihtsalt kustutama.
    - Kui lisate X-ile atribuudi nimetus, siis tuleb otsustada, kas laiendatud formaadis kasutusjuhtude mudelis (ja hiljem rakenduses) näidatakse nimetust X nimekirjades (kust saab valida tegevusi X-ga). Nendes nimekirjades võiksid olla sellised X-i andmed, mille alusel neid tüüpiliselt otsitakse. Neid andmeid ei tohiks olla liiga palju, sest muidu muutub nimekiri ebaülevaatlikuks. Soovitan järjekindluses ja kooskõla huvides näidata kõikides sellistes nimekirjades X-ide kohta samu andmeid.
14. CRUD maatriksi täiendamine.
    - Vaja on lisada uued Teie poolt lisatud olemitüübid ning määdata nende kasutus kasutusjuhtudes.
      - NB! *Klient* ja *Kliendi_seisundi_liik* andmeid loetakse kasutusjuhus *Tuvasta kasutaja*.

- Vaja on kirjeldada andmete kasutamine kasutusjuhtude 3 ja 6 käigus.

15. Kontseptuaalsest andmemudelist andmebaasi füüsilise disaini mudeli genereerimine, pidades silmas andmebaasisüsteemi, milles soovite andmebaasi realiseerida.
    - Kui teete projekti Enterprise Architecti abil, siis soovitan kindlasti kasutada parandatud mudeliteisendust, mille installeerimise ja kasutamise juhendi leiate: *Tarkvara saamine ja kasutamine => CASE: Enterprise Architect (ver 11) videod => EA mudeliteisenduse täiendus* See annab tulemuseks palju parema kvaliteediga (ja vähem muutmist nõudva) andmebaasi füüsilise disaini mudeli.
16. Füüsilise disaini andmemudeli parandamine.
17. Füüsilise disaini andmemudelist SQL lausete genereerimine.
18. SQL lausete käsitsi parandamine. Eriti puudutab see Enterprise Architecti kasutajaid. Lugege vajalike paranduste kohta kodulehelt järgmisest materjalist:
    - **Asukoht kataloogipuus**: */Iseseisva töö projekti tegemine /Juhendid* **Ressursi nimi**: Kontseptuaalsest andmemudelist andmebaasini (füüsilise andmebaasi disaini mudeli loomine; SQL koodi genereerimine, parandamine, käivitamine)
19. Genereeritud SQL lausete käivitamine valitud andmebaasisüsteemis.
20. Selliste kitsenduste jõustamine, mille kontrollimise realisatsiooni ei loodud käivitatud SQL lausete abil.
    - Kui teete projekti MS Accessis, siis *ülesanne 1* annab detailse juhendi, kuidas seal valideerimisreegleid jõustada. Jõustatavad reeglid leiate kontseptuaalsest andmemudelist.
21. Juhataja töökohale vastava rakenduse realiseerimine.
    - Realiseerida tuleb kõik juhataja kasutusjuhud, v.a *Tuvasta* kasutaja. Jätke prototüübi kasutajale mulje, et olete konkreetse juhatajana sisse loginud.
    - Leiate vajaliku rakenduse näite kodulehe kataloogist *Iseseisva töö projekti tegemine/Näiteprojektid/Ruumid (malli põhjal, osaline)*
22. Olemi-suhte diagrammide ja andmebaasi diagrammide dokumenti kopeerimine.
23. **Täita ülejäänud töövihiku dokumendi ossa jäänud tühjad/ülevaatamat kohad.**
24. Kirjeldada lisandunud kasutatud materjalid ja viidata neile töö tekstis: http://www.lib.ttu.ee/pdfs/viitamine.pdf
    - Kui Te seda ei tee, siis eksite infotehnoloogia teaduskonna õppuri akadeemiliste tavade rikkumise ja vääritu käitumise menetlemise korra vastu: https://www.ttu.ee/teaduskond/infotehnoloogia-teaduskond/it-tudengile/oppetoo-korraldus-13/vaaritu-kaitumise-menetlemise-kord-3/

Millele tuleb veel tähelepanu pöörata?

Loodud mudelid peavad olema hästi kasutatavad, et neid oleks lihtsam lugeda, edasiarendada, taaskasutada ja nende abil modelleeritava süsteemi probleemidele võimalikult varakult jälile saada. Tarkvaraarenduse tehised (nt mudelid ja lähtekood), mis on kehvasti esitatud, võivad küll öelda ja teha õigeid asju, kuid neid on raske kasutada. Öeldakse, et neis on *tehniline võlg*

(https://digi.lib.ttu.ee/i/?8865) ja neid iseloomustavad erinevad *halvad lõhnad* (lähtekoodi halbade lõhnade näited: https://sourcemaking.com/refactoring/smells). Tehniline võlg, nagu igasugune võlg, tuleb enamasti kunagi intressidega tagasi maksta. Tehnilise võlaga mudelite korral võib see näiteks tähendada lohakalt tehtud mudelite tõttu süsteemi vigade hilist avastamist ja suuri kulutusi nende parandamisele. Programmeerijad peavad looma puhast (mitte halvasti lõhnavat) koodi (*clean code*) ja modelleerijad peavad looma puhtaid (mitte halvasti lõhnavaid) mudeleid. Halvasti lõhnava koodi ja halvasti lõhnavate mudelite puhul on enamasti tegemist samade üldiste probleemide (dubleerimine, halb vormistus, erinevate abstraktsioonitasemete segamini kasutamine, järjekindlusetus jne) ilmingutega. Lugege mudelite halbade lõhnade kohta järgmisest allikast.

- Käosaar, E., 2017. *Lähtekoodi halbade lõhnade üldistamine süsteemianalüüsi mudelitele*. Bakalaureusetöö. TTÜ Tarkvarateaduse instituut [WWW] HYPERLINK "https://digi.lib.ttu.ee/i/?8865" https://digi.lib.ttu.ee/i/?8865 [Kasutatud 10.09.2018].