**Süsteemi üldine kirjeldus**

Süsteemi eesmärk on implementeerida dokumendis *Lisa 22 – Nouded\_proovitoole* välja toodud ärilised ja tehnilised nõuded.

Süsteemi üldine ülesehitus näeb välja järgmine. Äriloogilised kasutusjuhud ja abstraktsioonid infrastruktuursetele sõltuvustele paiknevad *Booking*-nimelises projektis. Äriloogika jaguneb erinevate domeeni teenuste ja domeeni mudelite vahel. Domeeni teenuse eesmärk on siduda kokku terviklikud kasutusjuhud, delegeerides osa tööst domeeni mudelitele ja lisades juurde äriloogilised õiguste kontrollid ehk millised kasutajad tohivad mida teha ja kellel on juurdepääs millistele andmetele.

Infrastruktuursetele sõltuvuste implementatsioonidele on loodud enda projektid, et vajadusel projekt, mis soovib neid kasutada, saab lisada viite ainult sellele implementatsioonile, mida vajab.

*Booking.Repositories.EntityFramework* projektis on implementatsioonid Booking projektis olevatele repositoride abstraktsoonidele. Andmebaasi talletamine on hetkel lahendatud *Entity Framework Core* raamistikuga. Soovi korral saab selle asendada mõne muuga, ilma, et peaks äriloogika kihis tegema muudatusi.

*Booking.IdGenerator.InMemory* projektis paiknevad implementatsioonid, kuidas luua domeeni mudelitele id-si mälus.

*Booking.Web.API projekt* on puhverrakendus, mille eesmärk on tagada juurdepääs kasutusjuhtudele üle *HTTP* protokolli. Selle, kuidas luuakse vajalikud eelandmed, leiab klassist *WebApplicationExtensions*. Rakendus kasutab *Swagger* ja *SwaggerUI* raamistikku, et luua klientrakendus API-ga suhtlemiseks. Rakenduse käivitamisel avaneb *SwaggerUI*, kus erinevad API “Endpoindid” on dokumenteeritud. Eraldi on välja toodud kirjeldus, kuidas logida sisse soovitud test kasutajaga.

Kuna proovitöö kirjelduses oli välja toodud, et sobib kasutada “InMemory” andmebaasi, siis peale rakenduse sulgemist kõik andmed kaovad ja järgmisel käivitamisel algab kõik puhtalt lehelt.