

Proovitöö ajakulude table ja analüüs

Ülesanne	Ajakulu	Kõige keerulisem osa	Kasutatud abivahendid
Kasutajasisendite muutmine valikvastusteks	20 min	-	-
Registreeringute kuvamine	1 h	Thymeleafi kasutamine html-is	TicketSystemApp*
Registreeringute kustutamine	2h	Thymleaf, mappingu loogika	TicketSystemApp*
Registreeringute muutmine	4h	Siin jäin suuresti kinni dialoogi lahenduse taha, ei teadnud kohe kas peaks javascripti või controller mappinguga seda lahendama. Put request andis alguses ühe thymeleafi näpuka pärast errori mida netist otsides oli keeruline leida.	https://github.com/Reinisozolins/crud*
Registreeringute otsimine	2h	Tahtsin tegelikult filtreerimise juba andmebaasist pärismise tasandil teha. Crud repository @Query ei töötanud päris nii nagu lootsin näidete põhjal saavutada. Kohe Java predicate-i kasutades oleks läinud umbes 15 min	-
Serveri poolne aegade kattuvuse kontroll	20 min	-	
Mapping ja navigeerimine	1h	Mappingu loogikas on raske midagi keerulist tuua, muutsin paar korda strateegiat ja sellega läks aega.	
Stiliseerimine, paigutamine	Kuni 30 min	-	

Grooming, refactoring	1h	-	
Dokumentatsioon	45 min		
Kokku	umbes 13h		

Analüüs

Mis oli lihtne?

Kõige lihtsam oli minu jaoks töö jaoks vajaminevate vahendite leidmine, controlleri loogika väljamõtlemine ning üldisem HTML/CSS kasutamine, DTO –d kasutasin esimest korda, aga selle loogikast oli lihtne aru saada. Suuresti aitas mind nende tegevuste juures minu varasem kogemus Spring Boot + H2 ülesehitusega, teadmised HTTP requestide osas ja lihtsalt kogemus Javaga.

Mis oli raske?

Thymeleaf – Vaatamata varasemale kogemusele valmistas mulle TL ikka kõige rohkem peavalu. Võiks ju arvata, et kogemus Vue-ga aitaks, aga Thymleafi süntaks pole just kõige ilusam minu arust. Sellest hoolimata sain sellega ikkagi lõpuks hakkama.

PUT request – Kuigi hiljem vaadates tundus PUT requesti lahendus väga sarnane DELETE requestiga, siis selle dialogi kuvamise loogika tegi selle ülesande parajalt raskemaks. Debuggimine polnud ka selles kohas minu tugevam külge, sest olin paar korda juba õigel teel aga mõtlesin ümber, arvates, et nii ei saa.

Search ja CrudRepository @Query –

Tahtsin otsingusõne järgi filtreerimise andmebaasi päringus lahendada, et ei peaks iga kord kõiki registreeringuid kuvama.

Kahjuks ei saanud ma lõpuks seda osa (suuresti ajapuuduse tõttu), usun, et umbes 1 h näidete ja dokumentatsiooni otsimist oleks mind lahenduseni viinud.

Mida oleksin tahtnud paremini/veel teha?

1)Nagu eespool mainitud @Query realisatsioon

2)Juhendi järgi oleks kuupäev olema nii sisestatav kui valitav. Minu mõte selle koha pealt oli see, et hiljem kuvatakse olemasolevad ajad andmebaasist eraldi tabelist.

3)Rohkem stiili, tulemuste leht oleks võinud asenduda samuti pop-up dialoogiga näiteks.