

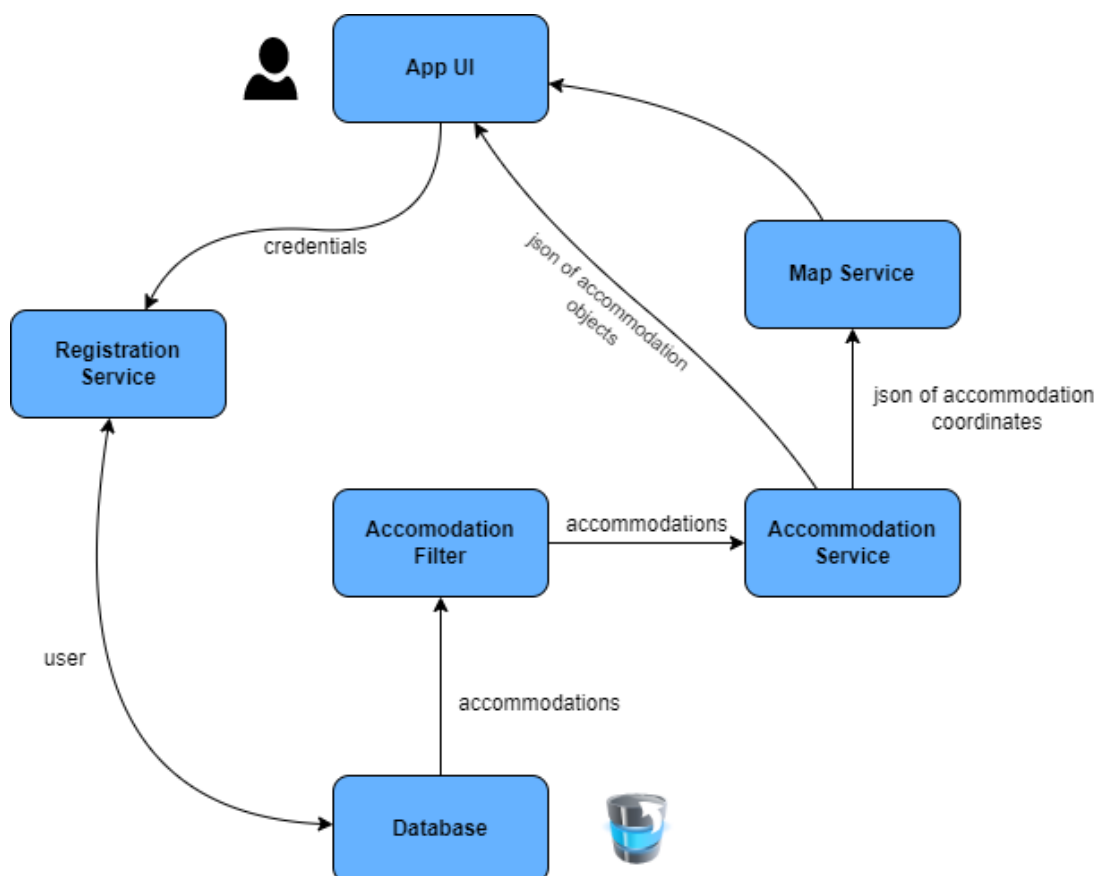
Архитектурни Погледи

Концептуална архитектура

Кога еден корисник сака да се регистрира или најави на системот, AppUI компонентата ги испраќа генералните на корисникот (во енкриптиран формат) до Registration Service компонентата која е одговорна да ги направи различните проверки за точноста на внесените податоци и да го зачува (или вчита) корисникот од базата на податоци.

Кога еден корисник сака да пребарува сместувачки објекти по различни пребарувачки критериуми, Accommodation Filter компонентата генерира соодветни прашалници до базата на податоци и така пронајдените објекти се препраќаат до Accommodation сервисот кој прави дополнителни проверки и ги трансформира податоците во формат соодветен за обработка на front-end. Уште повеќе, Accommodation сервисот комуницира со Map сервисот (имплементиран на front-end) чијашто задача е да ги претстави координатите на соодветните сместувачки објекти на мапа која ја гледа корисникот.

AppUI компонентата може да направи барање и за омилените сместувачки објекти на корисникот, но размената на информации меѓу компонентите е идентична како и претходно, со единствена разлика на прашалниците кои се генерираат за преземање на податоци од базата на податоци.

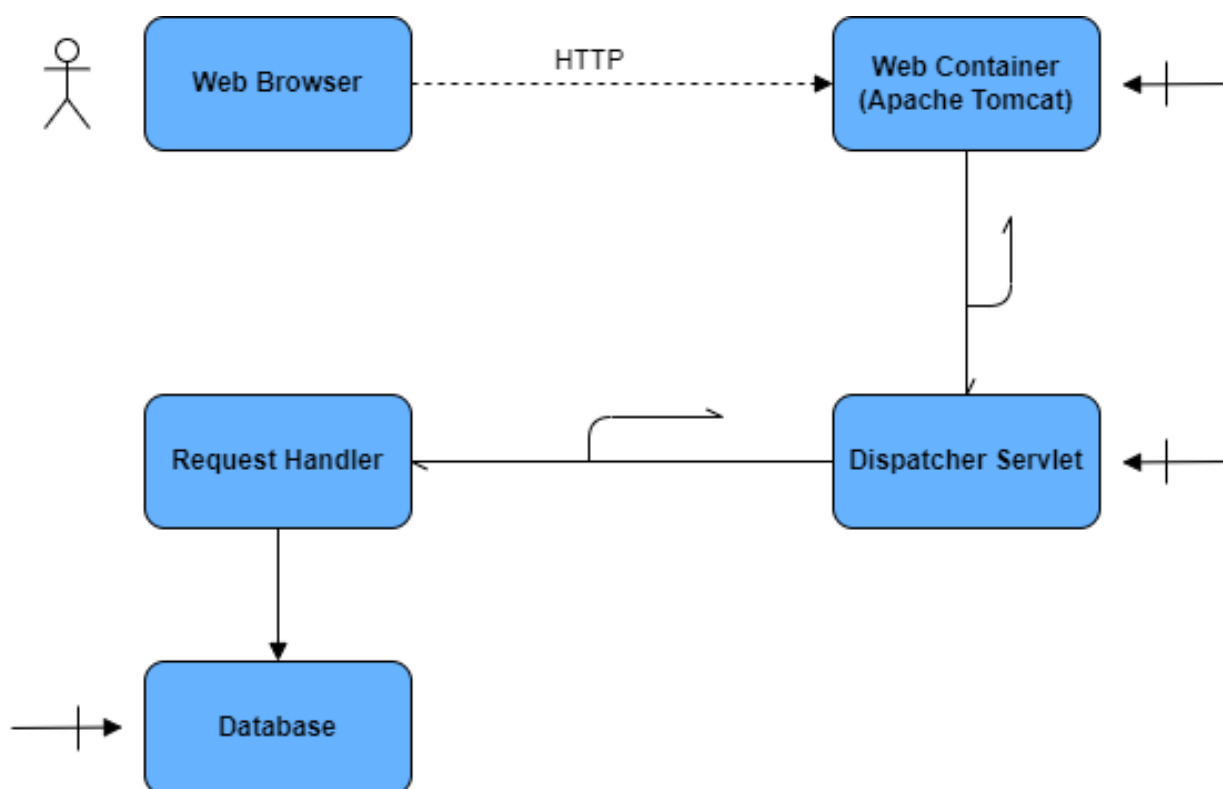


Извршна архитектура

Кога едно барање од клиентот пристигнува до веб серверот, Tomcat креира нов тред кој е одговорен да се справи со новодобиеното барање. Имено се испраќа асинхрон повик до Dispatcher Servlet компонентата, со цел Tomcat да може да продолжи да опслужува останати барања (без да чека едно барање да биде целосно обработено). Конекторот е од типот callback бидејќи иако повикот се извршува асинхроно, Tomcat компонентата сепак очекува да добие одговор од диспачер компонентата. Од тие причини, Tomcat компонентата припаѓа на стереотиот *service*.

Препратеното барање пристигнува до главниот диспачер сервлет кој е одговорен да го пренасочи истото до соодветниот хендлер, при што повторно имаме асинхрон повик (callback конектор).

На крај, останува уште комуникацијата со базата на податоци (која исто така е компонента од стереотипот *service*), но во овој случај повикот е синхрон, бидејќи процесот не може да продолжи да се извршува се додека не ги добие потребните податоци од база.



Имплементациска архитектура

Делот за front-end во имплементациската архитектура на нашиот систем ги користи апликациските компоненти: *React* и *React-Leaflet*, кои се извршуваат на самиот web browser. Со помош на *Axios* испраќаме барања до backend серверот. *Axios* користи HTTP барање преку кое е овозможена комуникацијата со backend серверот.

Како веб сервер го користиме *Tomcat* кој е веќе овозможен од *Spring Boot*. Најпрво секое барање од клиентот е преземено од презентациски слој, кој цели кон справување со истото. Имено секој од handler-методите во Rest контролерите се однесуваат на некое возможно барање од клиентот. Од друга страна имаме двонасочна врска кон базата на податоци за читање и пишување преку Repository слојот. Логиката за справување со барањето се врши во Service слојот и враќа резултат кон контролерот.

