

Documentatieverslag Webtech Opdracht 1



Door: Max Vasterd, Laurens Martos

Klas: EIN2Ve

Docent: Harm Frielink

Inhoudsopgave

- Inleiding blz 3.
- Domein Model blz 4.
- Verantwoording blz 5,6.
- Testplan blz 7.

Inleiding:

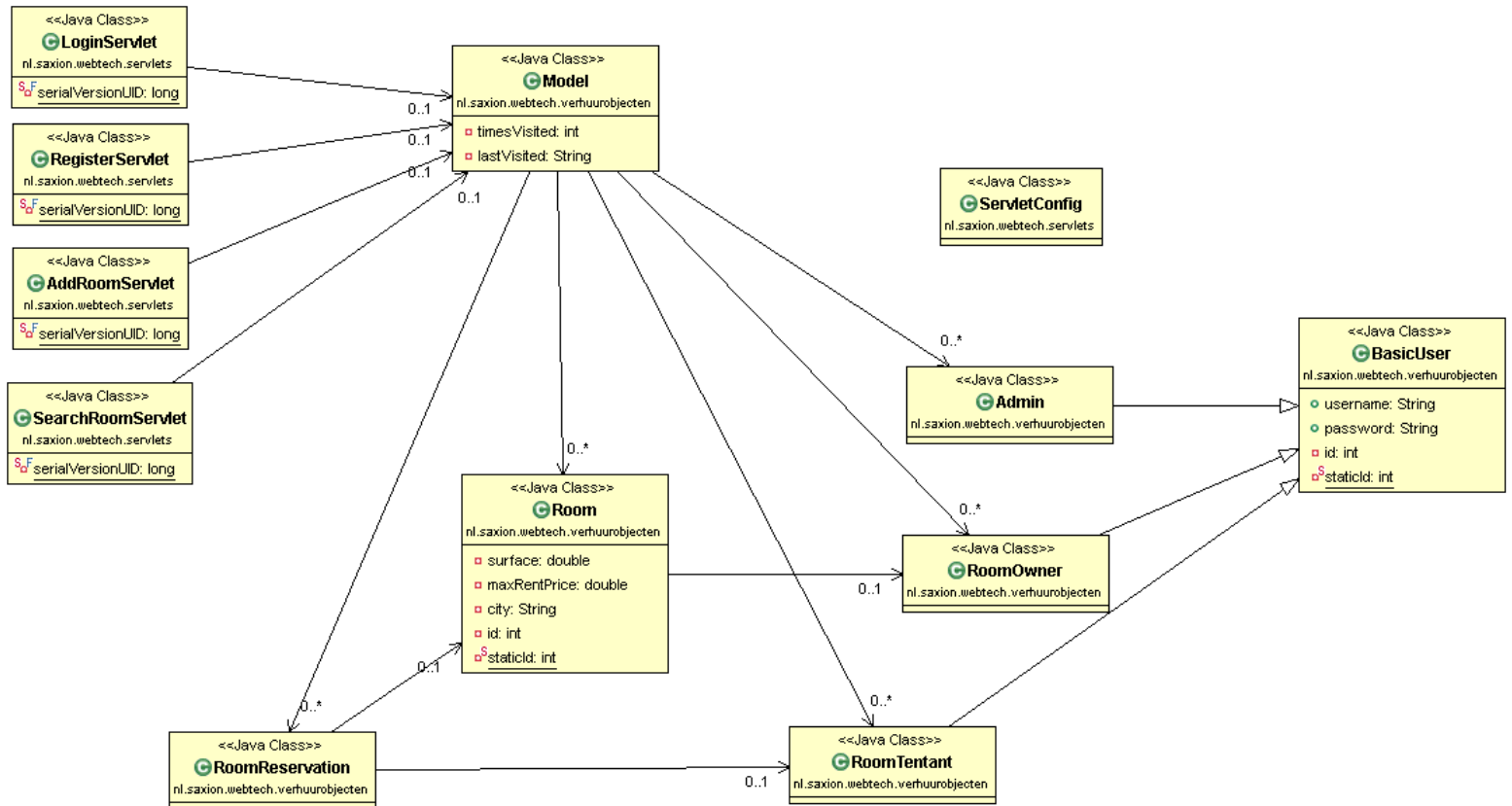
In dit document zullen we de toelichting geven tot de servlet applicatie die wij hebben gebouwd voor opdracht 1 van webtech, deze opdracht hield in dat wij een server-side applicatie gingen maken op basis van html, java en servlets waarin de volgende mogelijkheden uitgewerkt werden.

- Gebruikers moeten via een registreer pagina een nieuw account kunnen aanmaken en hierbij aangeven of zij een huurder of verhuurder zijn.
- Gebruikers moeten kunnen inloggen met de accounts waarmee zij geregistreerd zijn.
- Wanneer een gebruiker foutief inlogt moet een error pagina weergegeven worden.
- Verhuurders moeten kamers kunnen aanmaken op basis van oppervlakte, prijs en locatie.
- Huurders moeten kamers kunnen zoeken door middel van het opgeven van oppervlakte, prijs en locatie.
- Administrators moeten een lijst kunnen zien van alle geregistreerde gebruikers.
- Administrators moeten op de pagina met de lijst van gebruikers kunnen zien hoe vaak deze pagina is bekeken en wanneer deze voor het laatst is bekeken.

Domein model:

Voor dit project hebben wij een domein model opgesteld die de relaties laat zien tussen de classen en servlets die wij gemaakt hebben.

Dit domein model is hieronder te vinden.



Mocht dit domein model niet goed leesbaar zijn kunt u deze als .png bestand vinden dat samen met de rest van het project is ingeleverd onder de naam "domain model.png".

Verantwoording:

In dit gedeelte van het document zullen wij elk onderdeel van het project zoals gedicteerd in de opdracht langsgaan en onze implementatie hiervan bespreken.

Servlets:

Onze applicatie maakt gebruik van servlets voor alle “actieve” componenten van onze applicatie, als voorbeeld nemen wij de login servlet. Deze servlet gebruikt het `init()` component om het model op te halen uit de `ServletContext` om een instantie te verkrijgen van het model die over de gehele applicatie hetzelfde blijft. De login servlet gebruikt het `doPost()` component om te bepalen of de gebruiker die op de submit knop drukt wel de juiste informatie gegeven heeft en hem dan doorstuurt naar de correcte pagina. Een voorbeeld van het gebruik van het `doGet()` component in onze applicatie is de `SearchRoom` servlet deze gebruikt het `doGet()` component om een webpagina te laten genereren wanneer een ingelogde huurder op de knop zoek drukt.

HTTPRequests en HTTPResponse headers:

Onze applicatie maakt gebruik van de request en response headers voor een aantal functionaliteiten in de applicatie een goed voorbeeld hiervan is wederom de login servlet. In de login servlet maken wij gebruik van de request header om de gebruikersnaam en het wachtwoord van de gebruiker die op submit drukt op te halen en deze vervolgens te controleren. In het geval dat een administrator inlogt zal de login servlet een nieuwe web pagina opstellen met de benodigde informatie en deze in de response header terug sturen naar de client.

ServletContext:

Wij maken in onze applicatie gebruik van de `ServletContext` om het model door te geven aan alle servlets die wij gebruiken dit is handig omdat wij zo het model gelijk kunnen houden over de gehele applicatie.

Cookie:

Wij maken gebruik van een cookie om een tijd op te slaan in de browser, dit gebeurt alleen wanneer een administrator inlogt en wordt gebruikt om de administrator te kunnen laten zien wanneer hij voor het laatst heeft ingelogd.

WEB-INF:

Wij gebruiken de WEB-INF folder om al onze webpagina's behalve de login en registreer pagina op te slaan zodat de pagina's waarvoor een gebruiker al ingelogd voor moet zijn om te bekijken ook daadwerkelijk ontoegankelijk zijn voor gebruikers die naar deze pagina's proberen te browsen zonder in te loggen.

Forward() en redirect():

Wij hebben ervoor gekozen om voor de meeste gevallen gebruik te maken van forward omdat zo de pagina's ontoegankelijk zijn voor mensen die niet ingelogd zijn. Een voorbeeld hiervan is huurder.html deze webpagina is enkel te bereiken wanneer een gebruiker succesvol is ingelogd als huurder waarna de login servlet de gebruiker forward naar huurder.html. Wij maken wel gebruik van een redirect na het registreren om de gebruiker terug te sturen naar de login pagina aangezien dit gewoon een publieke pagina is.

Web.xml

Wij maken enkel gebruik van de web.xml file om aan te geven dat login.html de hoofdpagina van de applicatie is zodat een browser hiernaartoe gestuurd wordt wanneer een gebruiker naar de website browsert zonder een specifieke html pagina aan te geven in de url.

Testplan