

CDIO del 3 - Rapport

02324 Advanced programming

Deltagere:

Christin Holt,	s073653	Emil Eriksen	s120561
Jens Nielsen	s123115	Khan Mohammad Noori	s122997
Kim Rylund	s123667	Magnus Vestergaard	s113727
Mathias Larsen	s113734		

Lærer:

Mads Nyborg

Indledning

I denne aflevering skulle laves et forholdsvis lille program, som skulle vise en recept-formular som patienter kunne udfylde online. Der er på forhånd givet nogle krav til selve recepten. Når recepten er indtastet, skal brugen gerne sendes videre til en mere forklarende side, som enten skriver at recepten er oprettet succesfuldt, eller om der evt. mangler noget.

Krav

Vi skal lave et web-interface hvor brugeren kan oprette en ny recept. Opskriften består af følgende data:

Navn	Datatype
Recept nr.	Heltal [1:99999999]
Vare nr.	Heltal [1:99999999]
Varenavn	Tekst (2-20 karakterer)
Nomial nettovægt (Den ønskede/optimale vægt)	Vægt (50-6000 gram)
Tolerance på netto i procent	Procent (0,1-10%), dog mindst svarende til 1 gram

Oprettelsesdato samt tidspunkt (Genereres automatisk af systemet)	Tid og dato
--	-------------

Der foreslås at dette laves som tre websider:

- En formular til indtastning af ovenstående værdier.
- En side der præsenterer brugeren for eventuelle fejlbeskeder.
- En side der viser resultatet når alle indtastninger er valideret.

Der er givet noget af koden til de tre ovenstående sider for at hjælpe os i gang.

Implementation

Inputform.jsp:

I jsp dokumentets body starter hjemmesiden med at tjekke om metoden som man er kommet ind på hjemmesiden med er "POST", hvis det ikke er det, så bliver der formet nogle tekstbokse hvor man kan indsætte data i, og en knap som går ind på samme hjemmeside igen men denne gang vil man komme ind på hjemmesiden via. metoden post.

Når man kommer ind på hjemmesiden via. metoden post, så vil hjemmesiden starte med at loade de data man indtastede tidligere, og derefter tjekke om alle de data man har indtastet indeholder fejl, som den i så fald gemmer som en lang tekststreng, hvor hver fejlmeddelelse vil være adskilt af et semikolon.

Hvis fejlmeddelelsen er tom, så vil brugeren blive sendt videre til adressen "VisOkData.jsp?" efterfulgt af nogle parametre svarende til de data man indtastede i hjemmesiden.

Hvis der var en eller flere fejl, så vil brugeren blive sendt videre til hjemmesiden "VisError.jsp?error_resultat=" efterfulgt af tekststrengen med fejlene i.

VisOkData.jsp og VisError.jsp:

VisOkData.jsp starter med at hive alle parametrene ud af adresselinjen hvorefter den printer noget tekst ud på skærmen, som bl.a. indeholder de data som var oppe i parametrene.

VisError.jsp hiver den ene tekststreng ud oppe fra parametrene, men behandler den ved at udskifte alle semikoloner med linjeskift før den printer fejlmeddelelsen ud på skærmen.

Test

Test er en meget vigtig del af udviklingen af software. Vi har derfor taget opgaven med at teste vores program seriøst for at kunne finde evt. fejl.

White box test

Gennem kodningen af programmet har vi udført white box tests for at sikre os at programmets interne dele virker som de skal. Disse tests har fungeret rimelig lavpraktisk, da vi ikke har kunne bruge JUnit testing på jsp-klasserne.

Black box test

Hen imod slutningen af kodningen for projektet har vi lavet en række black box tests for at undersøge om programmet understøtter den korrekte funktionalitet. En god måde at teste programmets funktionalitet er black box testing. For at gøre dette tester vi indtastningsværdierne i vores InputForm, både fordi det er den centrale funktion i programmet og fordi der ikke er så meget andet at teste.

De forskellige indtastningsfelter burde kunne modtage følgende input:

- Recept-nr: et heltal mellem 1 og 999999999
- Vare-nr: et heltal mellem 1 og 999999999
- Varenavn: en streng på 2 til 20 karakterer
- Nomial nettovægt: et heltal mellem 50 og 6000
- Tolerance: et tal mellem 0.1 og 10

Vores black box test afslører at varenavnet kan indeholde mellemrum alene, så længe at der er mellem 2 og 20. Derudover finder vi ud af at tolerancen ikke overholder kravene. Ellers efterlader vores "monkey test" vores program som et solidt program som ikke crasher, selvom der indtastes mange former for mærkelige ting ind i indtastningsfelterne.

Fejlen som blev fundet i tolerance er selvfølgelig blevet rette efterfølgende.

Konklusion

Brugeren kan nu oprette en ny recept på web-interfacet, som indeholder alle de data, der står i kraven. Web-interfacet er designet efter forsalget på 3 websider, en til indtastning af data, en til eventuelle fejlbeskeder og en side, der viser resultatet, når alle indtastninger er valideret.