

Monkey Business

- en webbshop.

Grupparbete i Web Applikationer (DAT 076), Chalmers tekniska högskola år 2012.

Grupp 12

Gustaf Werlinder

Markus Schützer

Martin Augustsson

Patrik Thituson

Generell information om projektet

Kurs: Web Applikationer (DAT 076), Chalmers tekniska högskola år 2012.

Gruppmedlemmar, grupp 12 (grupp WebApp12):

- Gustaf Werlinder
 - Namn på github: guswer
 - Födelsedata: 780606-xxxx
 - E-post: gustaf_werlinder@hotmail.com
 - Klass: DAI3
- Markus Schützer,
 - Namn på github: elefantbilen
 - Födelsedata: 840316-xxxx
 - E-post: markusschytzer@hotmail.com
 - Klass: DAI3
- Martin Augustsson
 - Namn på github: Mckan
 - Födelsedata: 910203-xxxx
 - E-post: maraugu@student.chalmers.se
 - Klass: DAI3
- Patrik Thituson,
 - Namn på github: Snadde
 - Födelsedata: 871129-xxxx
 - E-post: thituson@student.chalmers.se
 - Klass: DAI3

Projektet syftar till att skapa en enkel men fungerande webbshop. Grupp 12:s webbshop heter *Monkey Business*. Gruppen har använt Git som versionshanteringssystem och källkoden finns att betrakta på <https://github.com/Snadde/WebApp12>.

Systemöverblick

Monkey Business är en webbshop som är till för att sälja apor till kunder via Internet. Webbshoppen är byggd med Java Enterprise Edition 6 (JEE6) som plattform. Systemet bygger på *Maven* och kan för närvarande endast köras lokalt på en dator med hjälp av *Glassfish*.

Användare/Roller

Potentiella kunder

Potentiella kunder (alla som använder Internet) kan se shoppens offentliga sidor (produktkatalog, kundvagn, registrering).

Kunder

Kunder som registrerat sig i systemet kan utföra köp samt se och administrera privata uppgifter (användaruppgifter och köpordrar). I webbshoppen kallas en kund för *customer*.

Administratörer

Administratörer kan administrera produkthantering och kundhantering samt göra administratörer av kunder. I webbshoppen kallas en administratör för *admin*.

User stories/use cases

Följande user stories beskriver vilka behov webbshoppen ska möta.

Kunden ska kunna se kontaktinformation till företaget bakom webbshoppen.

Kunden ska kunna söka efter produkter och sortera träffarna.

Kunden ska kunna se hela produktlistan.

Kunden ska kunna se detaljinformation om en specifik produkt.

Kunden ska kunna lägga produkter i en kundvagn oavsett kunden är inloggad eller ej.

Kunden ska kunna se en enkel överblick av sin kundvagn.

Kunden ska lätt kunna upptäcka när förändringar sker i kundvagnen.

Kunden ska kunna ändra antalet av en produkt i sin kundvagn.

Kunden ska kunna ta bort produkter ur sin kundvagn.

Kunden ska kunna registrera sig i webbshoppen.

Registrerad kund ska kunna logga in.

Inloggad kund ska kunna logga ut.

Kunden ska lätt kunna se om den samme är inloggad eller inte.

Kunden ska som inloggad kunna köpa produkter.

Kunden ska som inloggad kunna se en lista över sina tidigare köp.

Kunden ska som inloggad kunna se detaljinformation om sina tidigare köp.

Kunden ska som inloggad kunna redigera sin profil.

Administratören ska kunna logga in.

Inloggad administratör ska kunna logga ut.

Administratören ska lätt kunna se om de samme är inloggad eller inte.

Administratören ska som inloggad som kunna se produktlistan.

Administratören ska som inloggad kunna lägga till produkter i web shopen.
Administratören ska som inloggad kunna förändra produktinformation.
Administratören ska som inloggad kunna ta bort produkter från web shopen.
Administratören ska som inloggad kunna se en lista över web shopens kunder och administratörer.
Administratören ska som inloggad kunna förändra användarinformation.
Administratören ska som inloggad kunna ta bort användare från web shopen.
Administratören ska som inloggad kunna göra en kund till en administratör.

Teknisk systemdesign

Tiers

I fallet med vår webbshop körs alla tiers på samma dator med hjälp av *Glassfish* som server. Vår avsikt har varit att implementera strikt MVC-struktur med hjälp av tre sorters Managed beans; *model beans*, *control beans* och *backing beans*. Det tog ett tag innan vi begrep hur det skulle hänga ihop så vi gjorde en del fel i inledningen av projektet och pga tidsbrist kan vi inte refaktorera och åtgärda de fall där vi avvikit från MVC-strukturen.

Client side presentation tier

De websidor som utgör webbshoppens UI som alstras av JSF:erna i *Server side presentation* tier samt en JEE 6 Client som i normala fall ska köras på en användares dator (i vår fall samma dator som hanterar övriga tiers).

Server side presentation tier

Backing beans och JSF-filer som körs på JEE 6 plattformen.

Server side business logic tier

Här finns våra *control beans* som sköter logiken i vår webbshop samt *model beans* som sköter kontakten med databasmodulen (vid namn Monkey Business DB). Körs på JEE 6 plattformen.

Server side integration tier

Vår databasmodul (vid namn Monkey Business DB) innehåller de javaklasser som tack vare JPA (vårt ORM-redskap) kan sparas i och hämtas från databasen. Körs på JEE 6 plattformen.

Enterprise information system tier

Vår JPQL-databas (vid namn MBDB).

Modulpaket

Source Packages

Backing Beans

De beans som tillsammans med xhtml-sidorna representerar vyn i vår modell. Varje xhtml-sida har en backing bean som innehåller informationen som sidan ska visa eller skicka vidare.

Control Beans

De beans som agerar som controller i vår modell och utför all logik. De sätter, hämtar och processar information från backing beans. De anropar model beans för att utföra CRUD-funktionalitet.

Model Beans

De beans som agerar som model i vår modell. De håller ingen egen data utan är bara kommunikationslänken mellan control beans och databasmodulen.

Web-pages

WEB-INF

Innehåller konfigurationen för webbapplikationen. Här definieras t.ex loginkonfigurationen, roller för användarna samt vad rollerna har tillträde till i webbapplikationen.

admins

Innehåller de sidor som endast en administratör ska ha tillgång till. Detta styrs genom JAAS. Här finns de sidor som kan skapa/redigera/ta bort produkter och kunder.

customers

Innehåller de sidor som endast kunderna ska ha tillgång till. Styrs genom JAAS. Vi har även valt att admin inte ska ha tillgång dessa sidor då en admin enligt vår åsikt inte ska kunna handla i vår webshop. Här finns de sidor som skapar och visar purchase orders. Vi har även valt att ge kunderna möjlighet att redigera sin profil.

jsf

Innehåller de sidor som alla ska ha tillgång till så att alla kan se alla produkter, söka efter produkt, lägga till produkt i kundvagnen, visa kundvagnen, registrera sig samt logga in.

resources

Här ligger våra resurser, css för layouten, bilder till webshopen samt javascript för applikationen.

MonkeyBusiness DB

Denna modul används av model beans och innehåller javaklasser som har koppling till databasen mha JPA. Klasserna är annoterade för att kunna skrivas i databasen och få en

korrekt vektor. De klasser som skriver in i databasen är subklasser till den abstrakta klassen *abstractDAO* för att undvika kodduplicering. Innehåller även tester för att kontrollera att allting fungerar. Vi har även skapat en test-klass som itererar databasen med produkter samt en admin för vår redovisning.

UML och klassbeskrivning

Detta innehåll återfinns i slutet av rapporten.

Använda teknologier

- JAAS används vid inloggning och kontroll så att rätt användare får rätt behörighet och därmed når rätt sidor.
- jQuery används då en vara läggs till eller tas bort ur kundvagn för att uppnå blinkande effekt.
- Vårt UI byggs av JSF-sidor
- Managed beans används för att transportera information mellan olika delar av applikationen. Detta i så hög grad som möjligt i överens MVC-modell
- CSS3 används i vissa fall för ett behagligt utseende, till exempel skuggeffekten på "loggan" och de rundade hörnen på templatén.

Avslutande reflektion

Starka sidor hos vår webbshop:

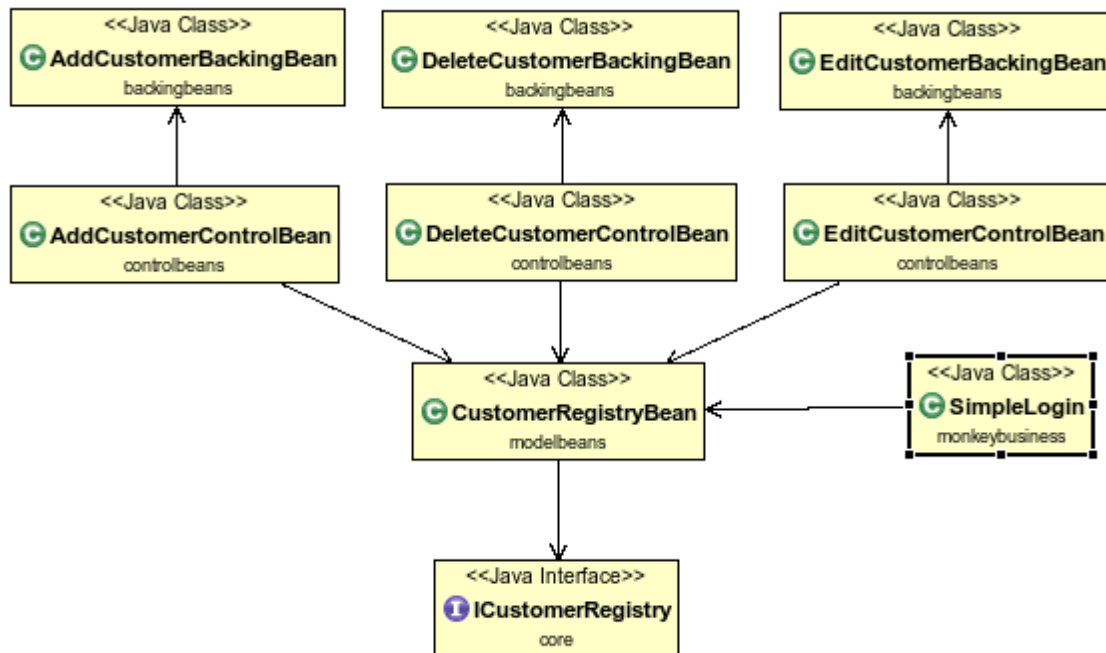
Över lag så är vi nöjda med vår webbshop. Vi tycker att den fungerar riktigt bra, särskilt med tanke på den begränsande tid vi haft till vårt förfogande. Särskilt stolta är vi över följande delar av vårt arbete:

- Vi har fått med många olika teknologier.
- Sidan känns funktionell, lättmanövrerad och trovärdig.
- Vi har lyckats implementera JAAS vid inloggning i systemet.
- Funktionell databas.

Vi är inte fullt så nöjda med:

- att vi inte hann med att refaktorera så att MVC-modellen följs till fullo.
- att vi inte hann med att skapa funktionen att admin kan administrera purchase orders.
- att vi varken hann med att få med HTML5- eller Prime Faces-teknologi, vilket hade varit roligt.

UML och klassbeskrivning



SimpleLogin

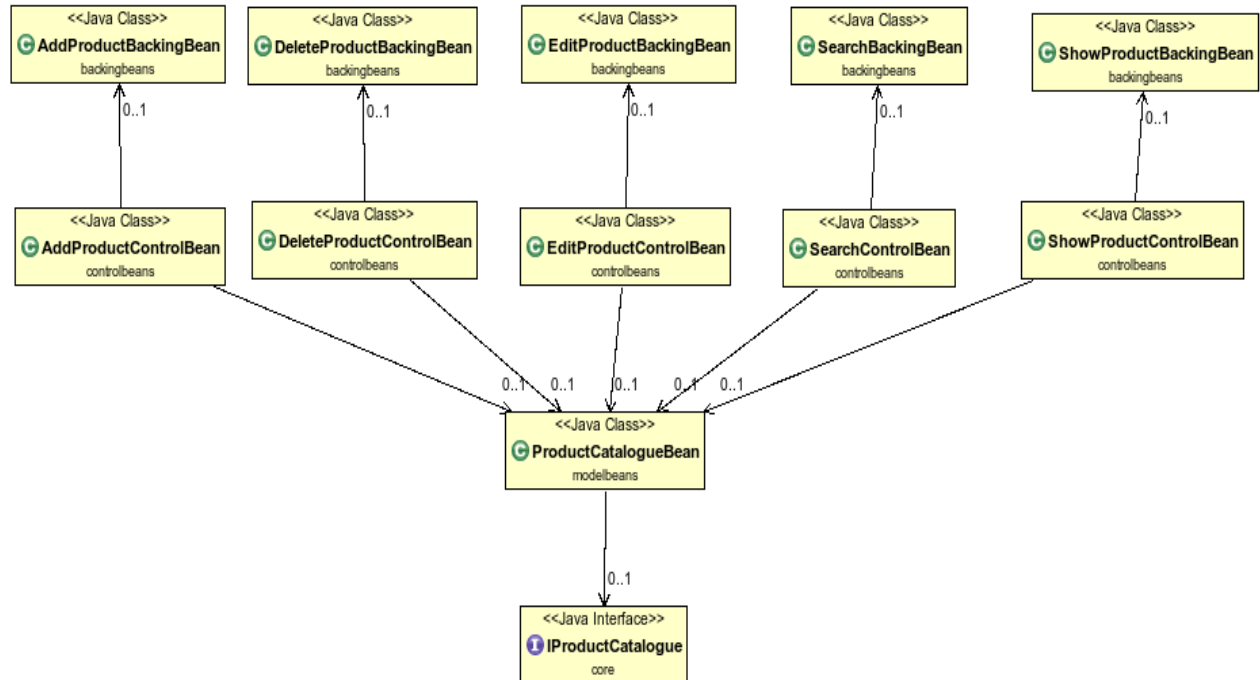
Hämtar den användaren som är inloggad för att kunna användas utav andra beans.

CustomerRegistryBean

Anropas utav control beans för att skapa/redigera/ta bort en customer ur databasen genom sin instance av CustomerRegistry

ControlBeans

Alla dessa beans använder sina Backing Beans för att sätta och hämta information som användaren vill göra och sedan utföra lämplig operation.

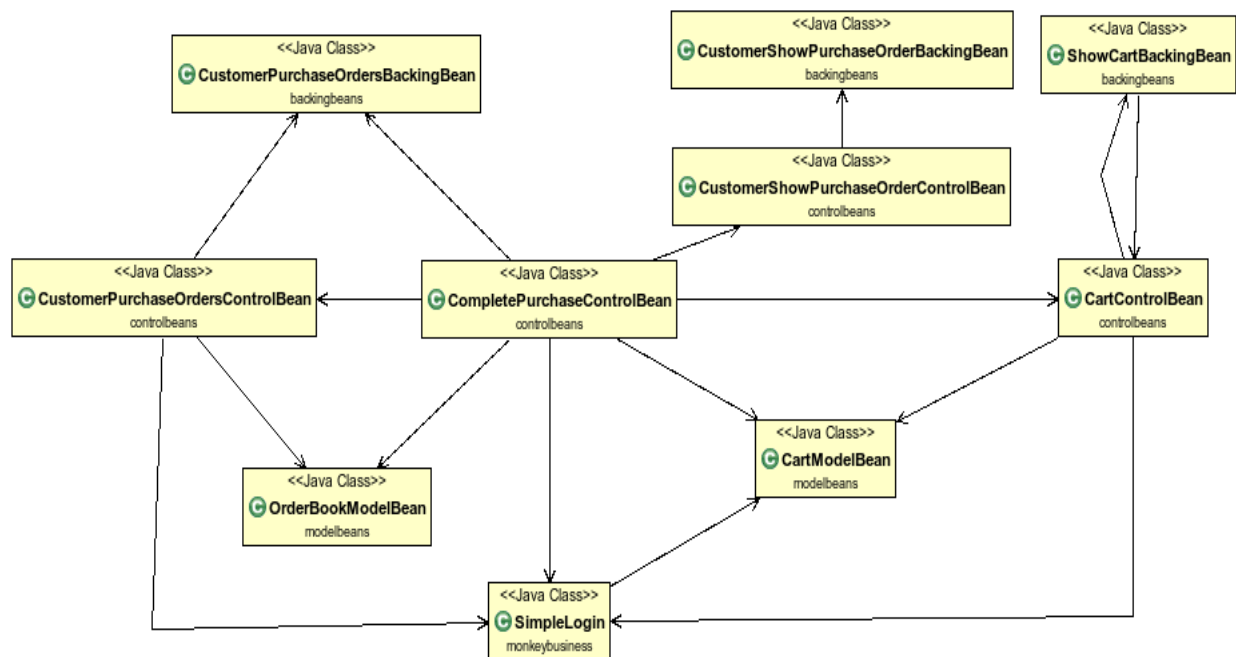


ProductCatalogueBean

Anropas utav control beans för att skapa/redigera/ta bort en produkt ur databasen, hämtar alla produkter eller de produkter man sökt efter genom sin instance av ProductCatalogue

ControlBeans

Alla dessa beans använder sina Backing Beans för att sätta och hämta information som användaren vill göra och sedan utföra lämplig operation.



CartModelBean

Denna håller alla produkter användaren lagt till i sin kundvagn.

CartControlBean

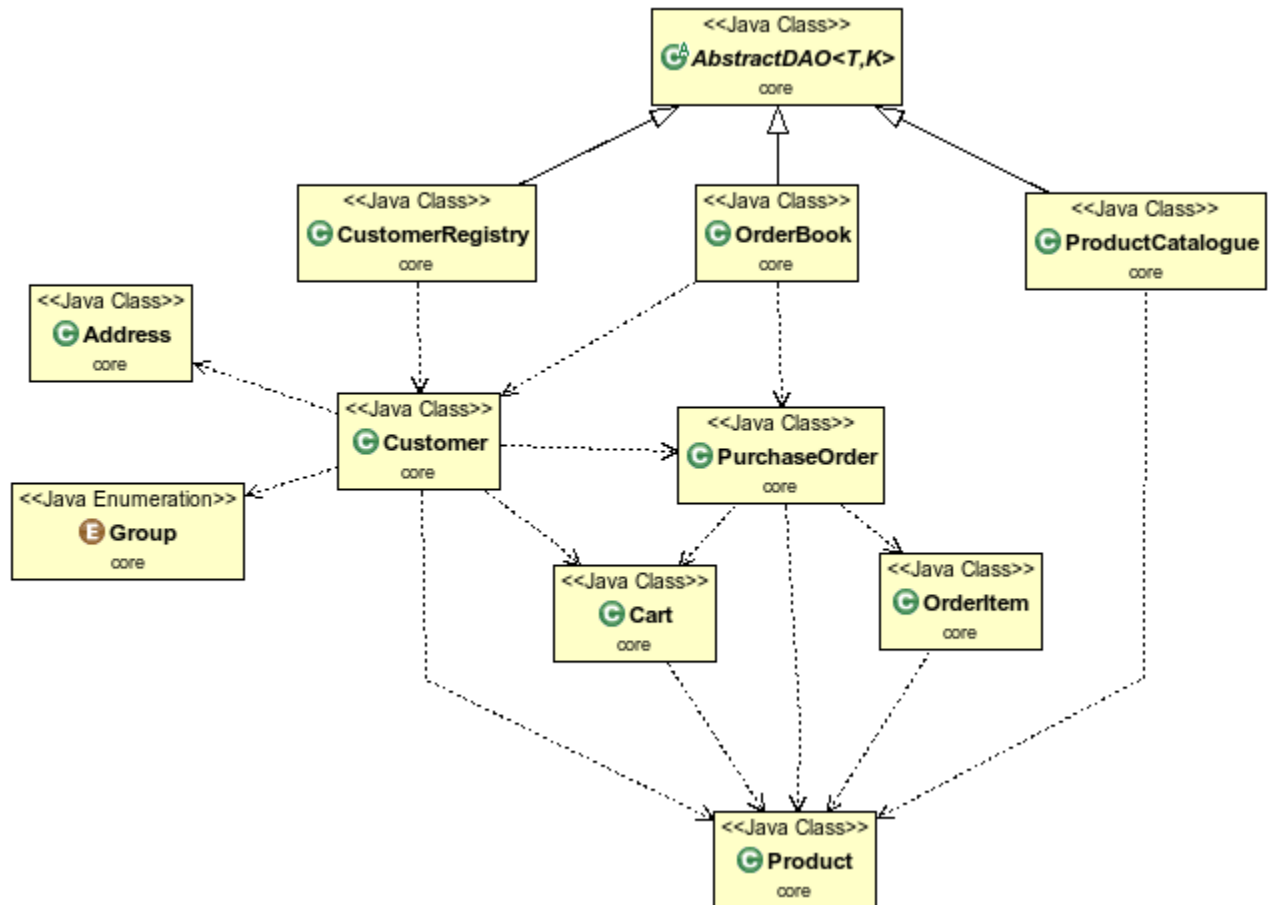
Lägger till produkter i CartModelBean när kunden lägger till produkter och uppdaterar ShowCartBackingBean

CompletePurchaseControlBean

Hämtar den inloggade användaren och knyter kundvagnen från CartModelbean till användaren. Skapar en purchaseorder om carten ej är tom och sparar den genom att anropar OrderBookModelBean.

SimpleLogin

Kontakts av Control beans för att få den inloggade kunden pga att man måste vara inloggad för att kunna slutföra köp.

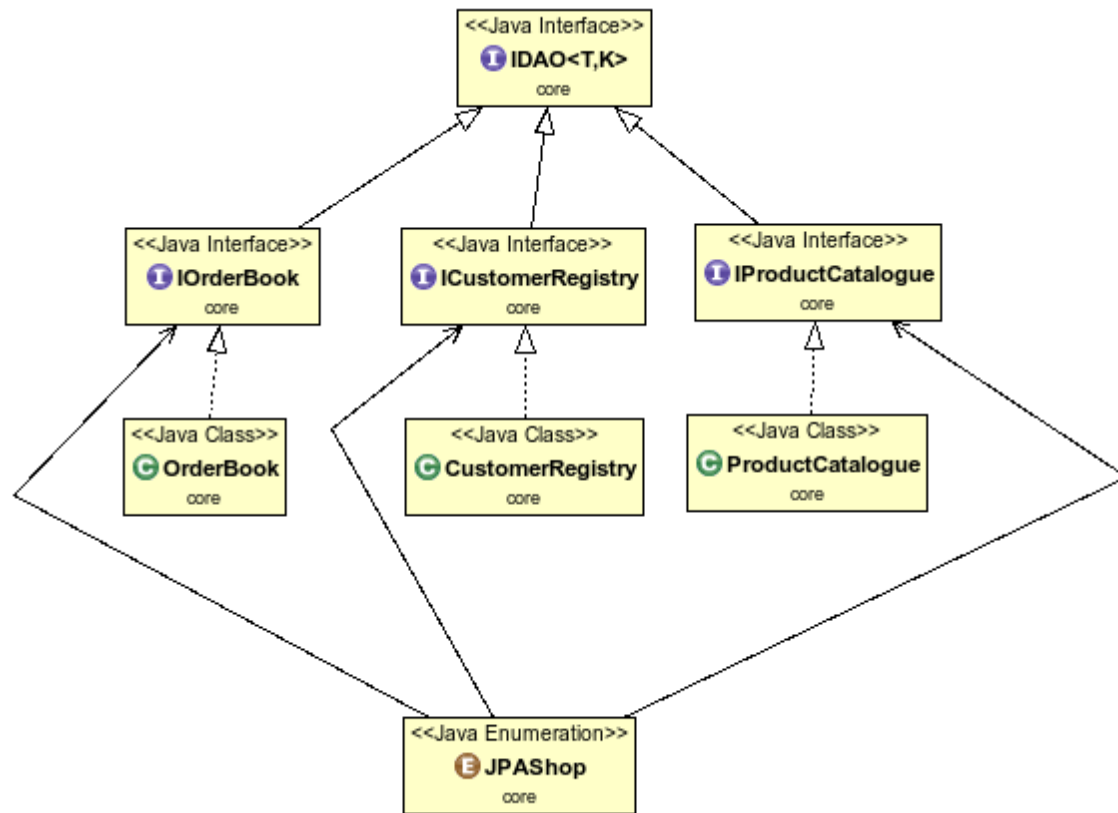


AbstractDAO<T,K>

Denna astrakta klass är super klass och innehåller generiska metoder för att kommunicera med databasen. Denna klass har full CRUD funktionalitet.

Group

Denna klass innehåller enum för att vi ska kunna implementera roller i applikationen.



JPAShop

Innehåller statiska metoder för att vi ska kunna få tillgång till **OrderBook**, **CustomerRegistry** och **ProductCatalogue**.