I starten av oppgaven med en ide, en plan og mål om å bruke test orientert programmering. Jeg skrev et par tester og satt i gang med programmeringen. Når jeg kom til hvor testene skulle komme inn fant jeg fort ut at de ikke fungerte sånn jeg ville. Ettersom jeg hadde begrenset tid på oppgaven ga jeg opp på å skrive tester for den og brukte heller det grafiske grensesnittet til testing.

Ved starten av utviklingen bestemte jeg meg for å bruke singleton pattern på alle datebase access objectsa mine (DAO). Dette var for å hindre å lage nye objekter hver gang jeg ville spørre databasen om noe. Alle DAOsene har sine egne spørringer og inserts fin pusset for datatypen de skal håndtere. Som f.eks ProductDAO lager et objekt av klassen Item som jeg kan bruke seinere med all dataen til raden som ble querriet. Dette var for å unngå å måtte åpne og lukke connectionen til databasen hele tiden.

Dao objektene mine bruker og en singleton klasse (ConnectionDAO) for å får tilgang til databasen gjennom dens Connetion som har mer en nok Null checkers og lignende forsikringer. ConnectionDAO forsikrer seg også om at database filen (oblig3.sqlite) finnes før den prøver å lage en connection mellom applikasjonen og databasen. Ved åpning av connection gjør den en rask spørring om den ser ut som den skal, hvis ikke inserter den sql skjemaet vedlagt.

# Hvilken designmaler har du brukt?

Av Creational design patters brukte jeg:

Singleton pattern – Alle Database object access er singletons.

Fra Structural Design Patterns har jeg brukt hint av

Fascade pattern og bridge pattern for å abstrahere vekk mye av database spørringene.

Og fra behavioral design patterns har vi pyntet med litt Template method patterns, mye i henhold til å åpne og lukke connections i rett rekkefølge. Jeg har også brukt en del state patterns for å oppdatere og forandre på UI en min, som database vinduet og fakturaen.

# Hvordan har du brukt dem?

Jeg har brukt disse design malene lett, og noe blandet etter som jeg var etter mange av fordelene med dem. Men jeg følte ikke at jeg trengte å implementere det for hele applikasjonen

# Hvorfor har du brukt dem?

Mange av dem har jeg brukt for å gjøre programmeringen mye lettere, når jeg abraherte vekk database spørringene og connection checks f.eks ble det mye letter å bare kalle de metodene singletonene mine hadde.

# Hvordan er de nyttig?

De er nyttige på den måten at jeg slipper å måtte tenke veldig teknisk for alt, hvis jeg har gjort den vanskelige delen av kodingen, hvorfor gjøre det igjen.

# Er det andre designmaler du kunne ha brukt som kunne gjort utviklingen enklere?

Jeg kunne brukt maler som Memento for å lagre database staten istedet for å requeste en på nytt hver gang jeg bytter / oppdaterer den. Så kunne jeg brukt mye mer av malene som facade og bridge for å gjøre koden mye mindre tettkobla.