# Brugervejledning til programmet

For at anvende programmet er der nogle få ting der skal gøres klar. Denne vejledning vil hjælpe dig til at nå til et punkt i programmet hvor der kan testes på det hidtil implementerede.

## Database opsætning

Sørg her for at der kører en fuld funktionel MySQL-server på maskinen som er klar til brug. Navigér herfra nu til klasse filen CSystemStart.java lokaliseret i programmets control-package. Her vil du i starten af klassen finde følgende stykke kode:

//Set databaseinformation

System.out.println(" - Setting up database..");

ADBInfo ***dbinfo*** = **new** ADBInfo();

dbinfo.setUser("root");

dbinfo.setPassword("123");

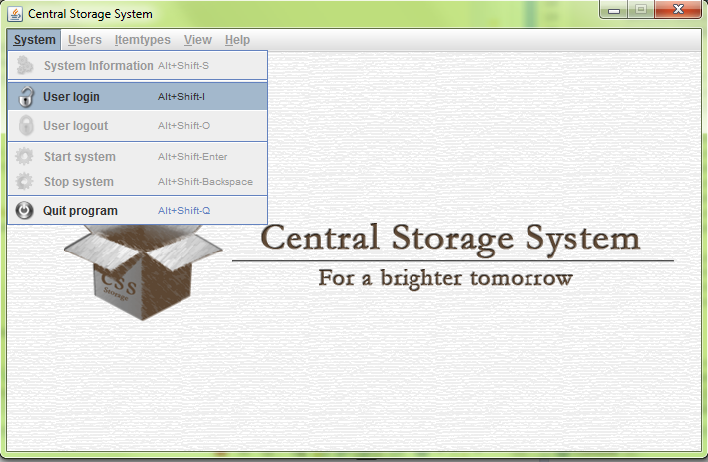
broker.setDB(dbinfo);

I metoderne setUser() og setPassword() erstattes parametrene "root" og "123"med henholdsvis din MySQL-servers brugernavn og adgangskode.

## Initialisering af programmet samt login

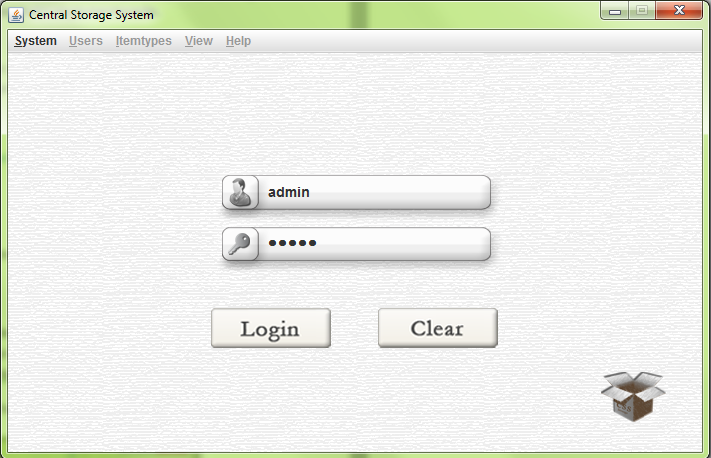
Da database forbindelsen nu er sat op og er klar til at blive forbundet til kan programmet startes. Dette kan enten ske ved at åbne projektet i en gyldig editor og kompilere koden vha. main-metoden lokaliseret i PGeneral.java eller ved at åbne den eksekvérbare .jar fil i projektmappen.

For herefter at få adgang til systemet skal man logge ind fra følgende menu:



Som standard er både brugernavn og password hardcodet til ”admin”. (Note: bruger-systemet er ikke implenteret, så denne login-form er kun for visuelle årsager til illustration af en eventuel login)

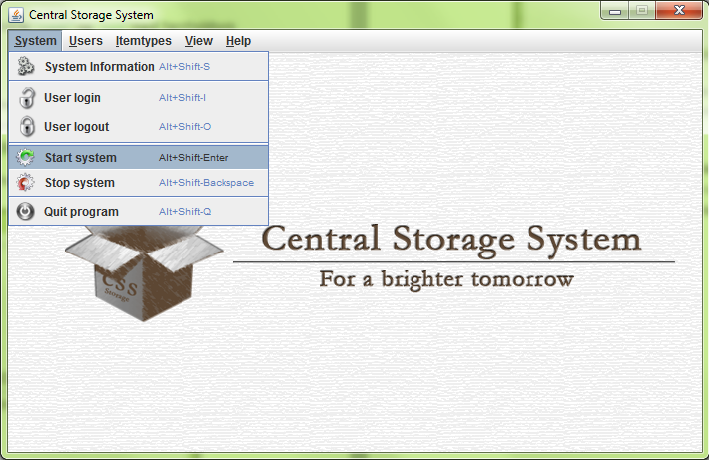
Et eksempel på login kan ses nedenfor:



Når login er blevet godkendt vil der blive åbnet op for diverse muligheder i menuen(bemærk her at meget få metoder rent faktisk er implementeret)

## Systemets opstarts rutine

Ved opstart af programmet bliver der læst en række af instanser ind direkte fra databasen idet at ”Start System” bliver triggered. Dette er en nødvendighed ved opstart for netop vores client (dette ville selvfølgelig blive gjort meget anderledes i den virkelig verden)



Da vi ikke har fuldført samtlige iterationer og implementeret samtlige usecases er der hertil også nogle instanser der er blevet hardcoded(f.eks. enkelte EItemTypes og EItems). Disse instanser bliver oprettet og gemt i databasen ved kald på System Start og vil derfor melde en fejl hvis man prøver at starte systemet en yderlig gang. (med flere iterationer skulle disse ikke hardcodes men snarere tilføjes via yderlige funktionaliteter)

## TCP Client

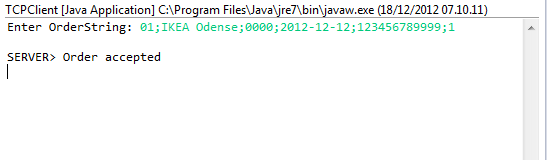
Ved initialisering af ’Start System’ rutinen bliver der oprettet en TCP server på en ny tråd. Denne server står og venter på en forbindelse fra en eller flere TCP klienter. (disse klienter repræsenterer de enkelte detail butikker som vil sende ordrer til vores lagersystem)

Yderligere udover vores centrale projekt har vi lavet en attrap som skal imitere den basale funktionalitet af en detail butik. Dette projekt kan også findes på CD’en navngivet ’DSSClient’.

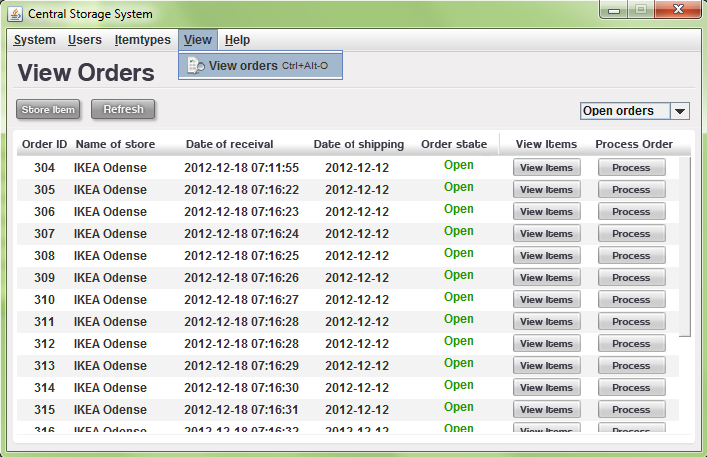
**Sørg for at serveren kører som tidligere nævnt** og kør så dette projekt. En prompt vil komme op hvor man kan sende en tekst-streng til serveren. Denne streng repræsenterer en ordre lavet hos en detailbutik.

**Eksempel-streng:** 01;IKEA Odense;0000;2012-12-12;123456789999;1

En mere detaljeret beskrivelse af denne protokol kan læses i rapporten under afsnittet ’Datakommunikation’.

****

En ordre bliver her oprettet og kan herefter ses i systemets View Orders menu.



Fra denne menu kan vores yderlige use-cases Process Order og Store Item så kaldes.