



# PROJEKTARBETE

DAT076

## Dokumentation

Detta dokument är en dokumentation för det projektarbete som genomfördes under september och oktober månad i kursen Web Applikationer (DAT076) vid Chalmers Tekniska Högskola

Grupp 2

Adam Clark, Dag Fridén, Fredrik Lundberg, Oscar Blomqvist

## Innehåll

Projektmedlemmar .....	1
Introduktion .....	2
Förutsättningar .....	3
Användarfall .....	3
Register .....	4
Login .....	5
Add items .....	6
Add list .....	7
Visa lista .....	8
Ta bort lista .....	9
Logout .....	10
Fysisk uppsättning .....	11
Komponenter .....	11
Maven .....	11
Servlet .....	11
REST .....	12
Java Script .....	12
Modellen .....	12
Frontend .....	12
Backend .....	12
Applikationslager .....	12
UML Diagram .....	13
.....	13
Centrala klasser .....	14

## Projektmedlemmar

Adam Clark

[adam.clark@xenit.se](mailto:adam.clark@xenit.se)

911129-1838

Dag Fridén	<a href="mailto:dag@friden.com">dag@friden.com</a>	910122-2355
Fredrik Lundberg	<a href="mailto:lundberf@student.chalmers.se">lundberf@student.chalmers.se</a>	881110-4937
Oscar Blomqvist	<a href="mailto:boscar@student.chalmers.se">boscar@student.chalmers.se</a>	880228-5034

# Översikt

## Introduktion

Denna applikation är skapad för att göra olika att-göra listor. Några exempel på detta är en inköpslista för mat, olika uppgifter man ska göra, kom-ihåg lista till en fest och så vidare.

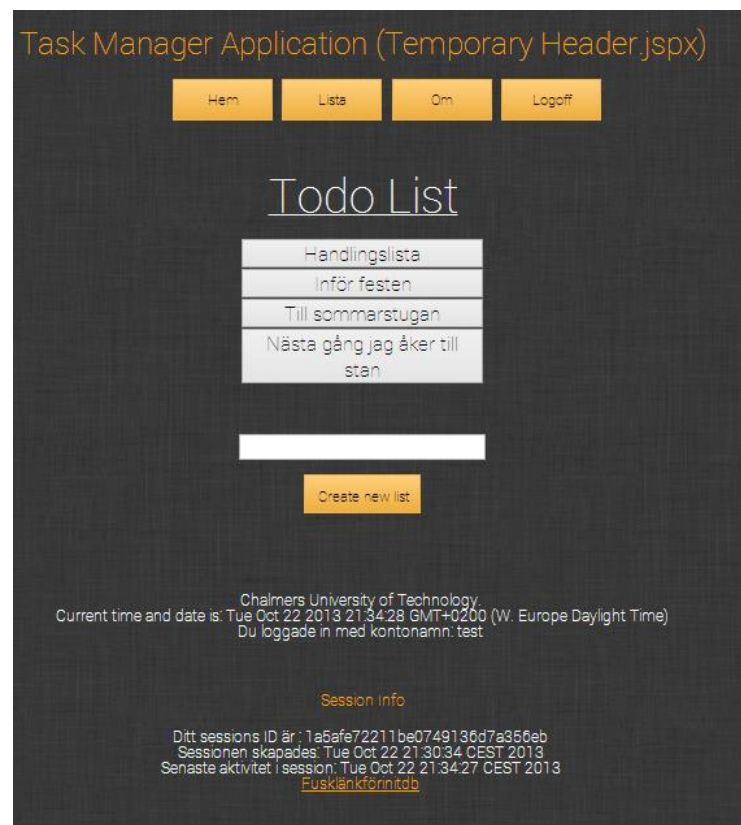
Hemsidan är uppbyggd på ett sätt som möjliggör inloggning med konto för att skapa privata listor vilka inte delas med övriga användare av applikationen. I en framtida vidareutveckling av applikationen kommer även delade och publika listor finnas med.

Listorna visas på ett intuitivt sätt på den webbsida där applikationen är publicerad.

För att vidare göra applikationen skalbar och enkel att hantera ligger det i bakgrunden en databas vars primära uppgift är att spara listor och inloggningsinformation.

Applikationen har även stöd för att vidareutvecklas till befintliga och nya användargränssnitt. Detta innebär till exempel att applikationen skulle kunna köra parallellt på en mobil enhet samtidigt som den används i en webbläsare – med samma information på båda platser.

Applikationen har utvecklats av fyra studenter vid Chalmers Tekniska Högskola under höstterminen 2013 i kursen Web Applikationer, DAT076.



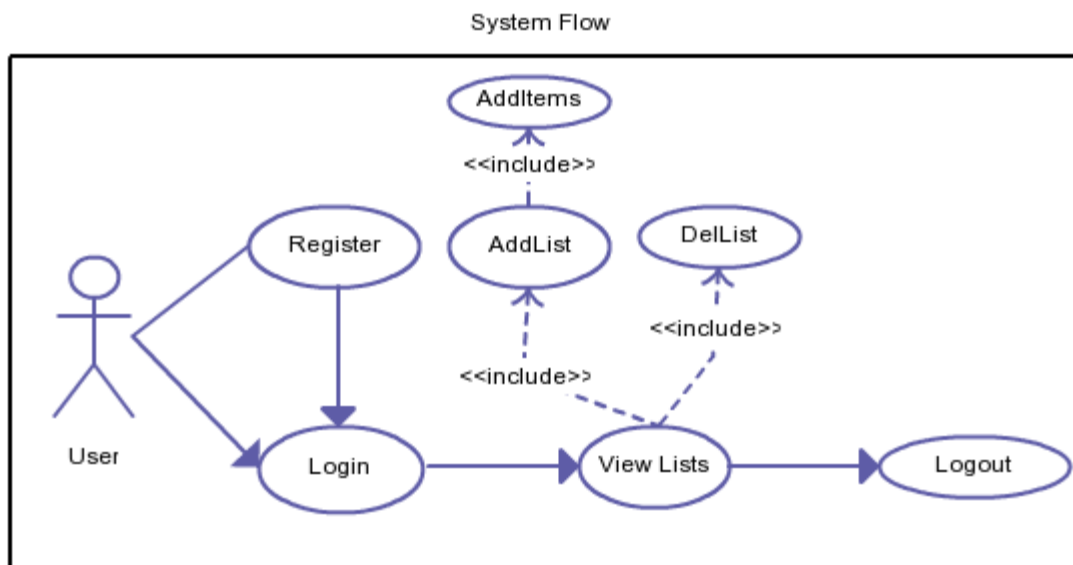
## Förutsättningar

Applikationen är modulärt uppbyggd med en så kallad frontend del samt en backend. Applikationen är byggd med Java Enterprise Edition och några av de övriga teknikerna/språken som använts är JavaScript (jQuery) och JavaDB (Derby).

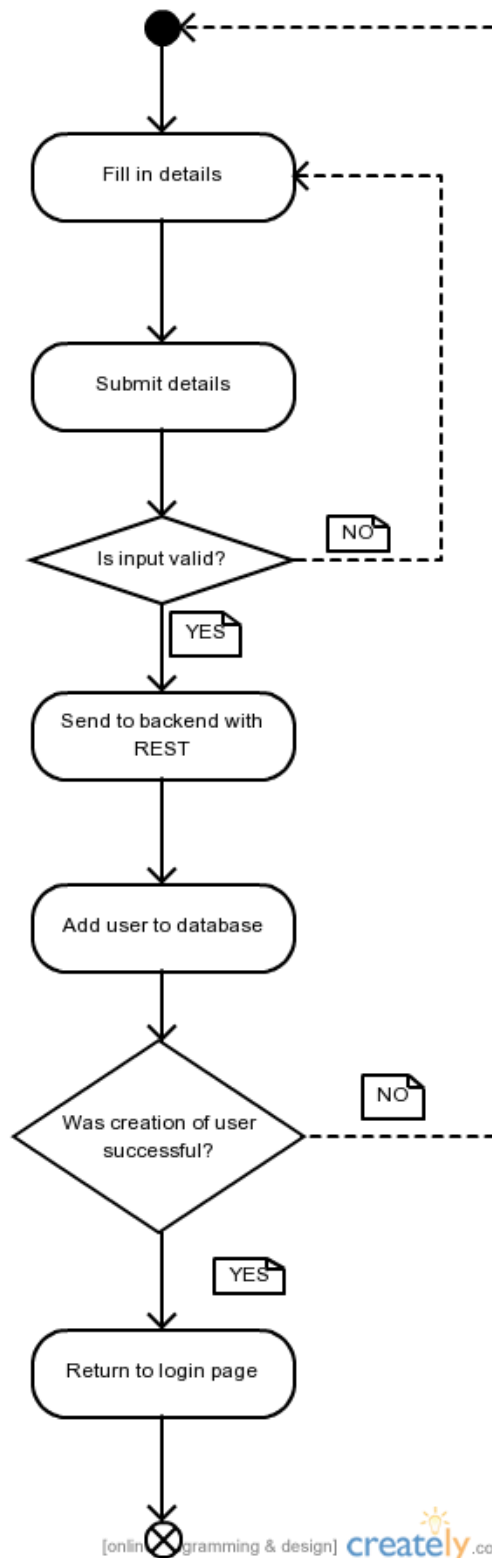
Under uppbyggnaden och testningen av applikationen är den körd på open-source applikationsservern Glassfish (CDDL/GPL).

## Användarfall

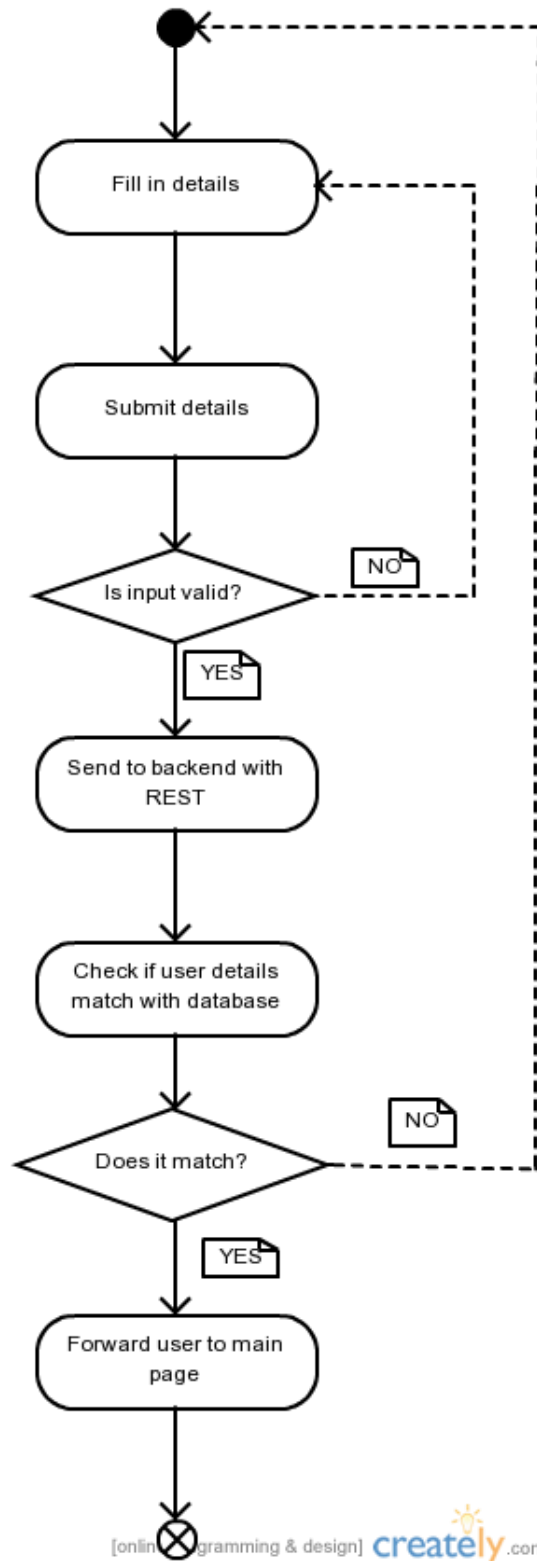
Nedan följer några av de användarfall som kan ske vid användandet av applikationen. Den första figuren förklarar olika user-case där varje cirkel innebär ett scenario vilket mer detaljerat förklaras i efterföljande figurer.



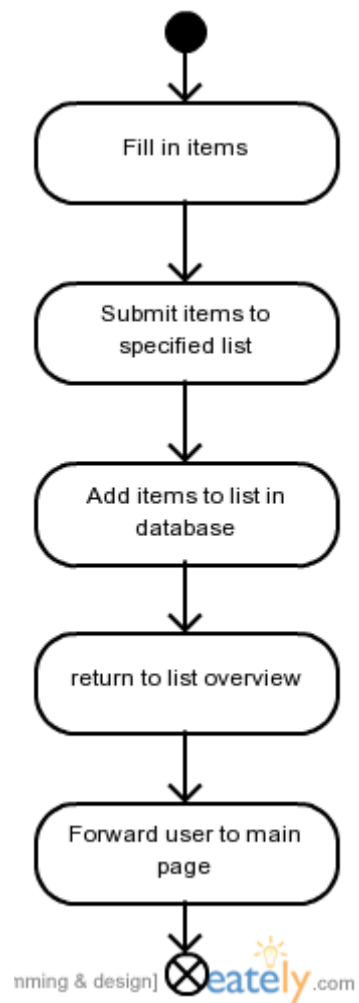
## Register



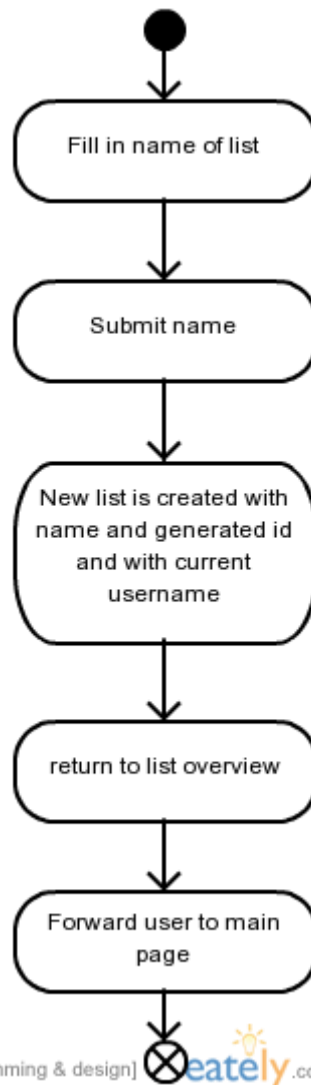
## Login



## Add items

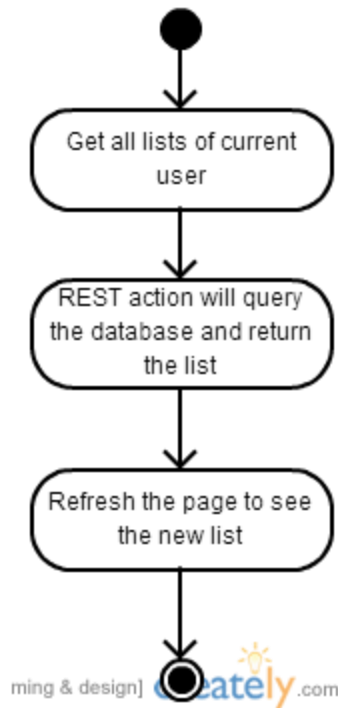


## Add list

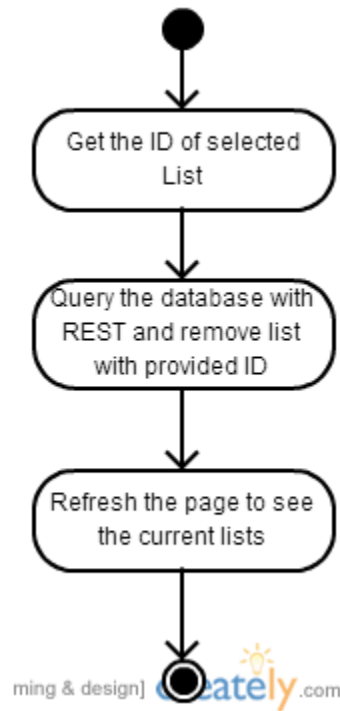




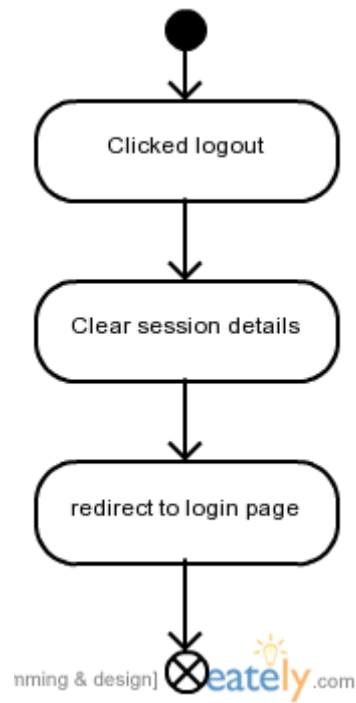
## Visa lista



Ta bort lista



Logout



# Teknisk Design

## Fysisk uppsättning

Vid en eventuell utrullning av denna applikation så skulle det krävas en webbserver som kan hantera Java EE applikationer. En väl fungerande server är den som användes vid utvecklingen av applikationen – GlassFish Server 3.1.2.2. Applikationen kräver en domän i Glassfish servern och inga speciella inställningar krävs för en grunduppsättning. Det finns många möjligheter att skala denna applikation med lastbalansering mellan olika servrar och till exempel lasta databasen mot ett databaskluster.

Det går att avlasta denna server genom att göra en migrering av databasen till en extern server. Detta är att rekommendera i det scenario då applikationen skulle rullas ut live.

Det finns komponenter i applikationen som inte är lämpliga att användas i en produktionsmiljö, dessa komponenter behöver designas om. Det rör primärt funktioner för att skicka data mellan olika vyer i applikationen samt hur användarnas session autentiseras.

## Komponenter

I denna applikation ingår många delkomponenter, de primära kommer i detta kapitel presenteras och en förklaring till i vilka delar av applikationen de används samt vilket syfte de används för.

Ett av projektmålen var att använda de olika teknikerna som används under laborationerna i kursen. Detta gav oss problem med att skicka variabler till JSPX sidor, som körs på servern, till JavaScript filer, som gör på servern.

### Maven

Maven är ett verktyg som används för att förenkla utveckling av Java-applikationer. Det används för att automatiskt bygga programmet och hämta de beroenden applikationen har. Maven använder sig av ett XML-dokument som heter POM.xml och det är i denna fil projektet beskrivs och vilka beroenden som applikationen har. Beroenden är olika paket och moduler som krävs för att applikationen ska fungera. Till exempel används Maven för att hämta de beroenden som krävs för att applikationen skall kunna arbeta mot en databas.

Maven används både i den frontend och backend applikationen består av.

### Servlet

Filen "TaskMgrServlet.java" innehåller funktioner för autentisering av användaren och identifiering av sessionen.

## REST

Filen "TaskManagerResource.java" innehåller funktionalitet för att kommunicera med modellen. Funktionerna görs genom att användarens webbläsare gör anrop mot i förväg konfigurerade URL:er . Funktionerna som körs anropar sedan "core"-objektet som ger tillgång till funktionerna som återfinns i Backend.

## Java Script

En stor del av hemsidan utgörs utav att hämta och skicka data som är associerad till användarens listor. Dessa funktioner utnyttjar AJAX och jQuery. Java script används även för att dynamiskt skapa listorna som visas upp för användaren.

## Modellen

Projektet är uppdelat i två projekt. Dessa två projekt, "Backend" och "Frontend", är skapade för att separera filer som hanterar det grafiska gränssnittet en användare utnyttjar och funktioner för att hantera logik och data.

## Frontend

Detta projekt innehåller fil och funktioner som hanterar de grafiska komponenter en användare ser i applikationen. Nedan finns en förteckning över de filer som återfinns i paketet fronten.

## Backend

I Backend finns funktioner som t.ex. sparar och hämtar data från databasen, krypterar lösenord och skapa databasen.

## Applikationslager

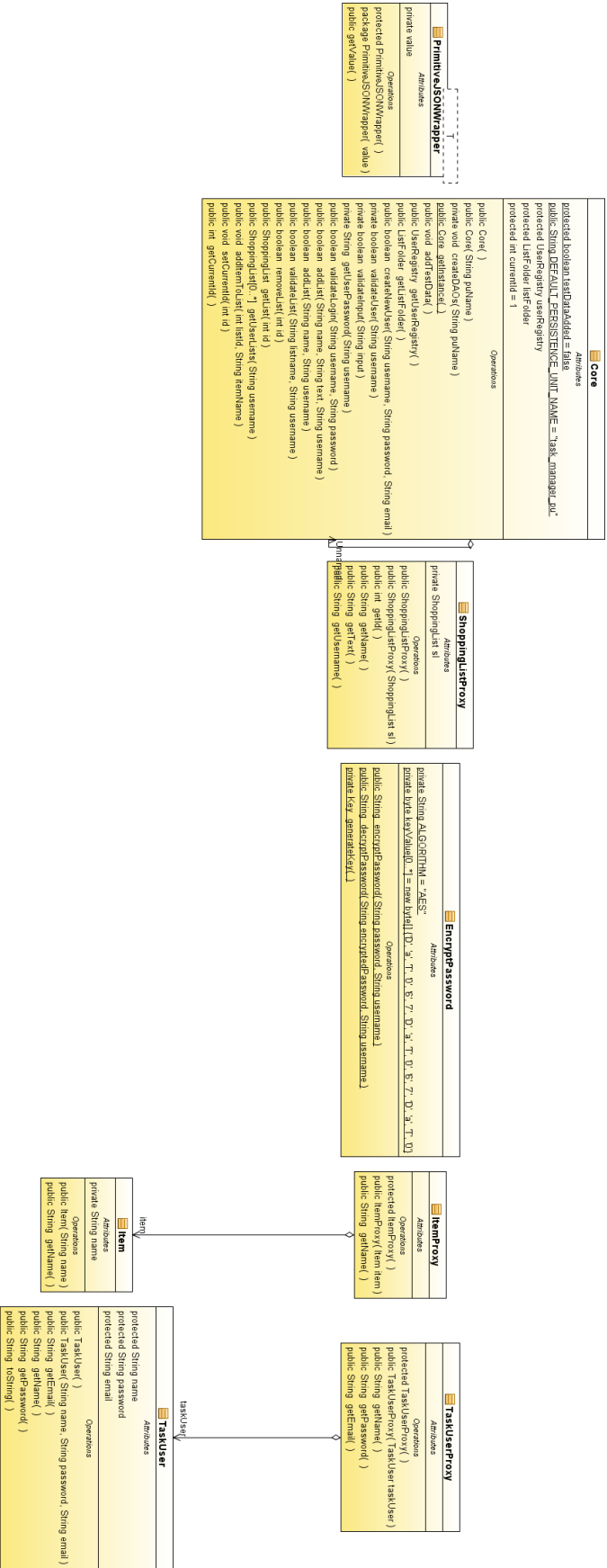
Applikationen kan delas upp i tre lager, klient server och data. I Klientlagret ligger filer som används för att skapa grafisk data på klienten. I Serverlagret ligger funktionen som behövs för att leverera innehåll (data) till klienten, och i det nedersta lagret finns data. I detta projekt har ingen "riktigt" databas används utan istället har en "Derby DB" skapats enbart för att kunna visa på funktionaliteten som en databas kan ge. Detta med motiveringen att inte ex. databasstruktur är betygsgrundande i denna kurs.

Klient: HTML CCS JS JSPX

Server:Servlet och REST

Data: Derby DB

UML Diagram



## Centrala klasser

Klassnamn	Funtion
Core	Applikationens "basklass". Alla funktioner som anropas från REST går via denna klass.
EncryptPassword	En klass som används för att kryptera lösenord. Vi använder här en AES metod som även kan används för att kryptera användarens data i databasen. Det vore mer lämpligt att göra en MD5SUM, men genom att implementera en funktion som krypterar all data som användaren sparar i systemet.
Item	Ett objekt som används för att representera varje rad i en lista.
ItemProxy	En proxy som används för att skapa JSON eller XML data av ett "item" objekt.
ListFolder	
ShoppingList	En objekt som representerar en lista med "items".
ShoppingListProxy	En proxy som används för att skapa JSON eller XML data av ett "ShoppingList" objekt.
TaskManagerResource	REST.
TaskUser	Ett objekt som används för att representera en användare.
TaskMgrServelet	Innehåller funktioner för att autentisera en användare.