

Matteus Laurent, Johan Levinsson, Oscar Petersson, Erik Peyronsson

## **MatLabb - Designspecifikation**

Högskoleingenjörsutbildning i datateknik, 180 hp

Designspecifikation - 1 oktober 2015  
**Programmeringsprojekt, HT15**  
TDDI02, Linköpings universitet

Handledare:  
Johan Frimodig  
Institutionen för datavetenskap



# Innehåll

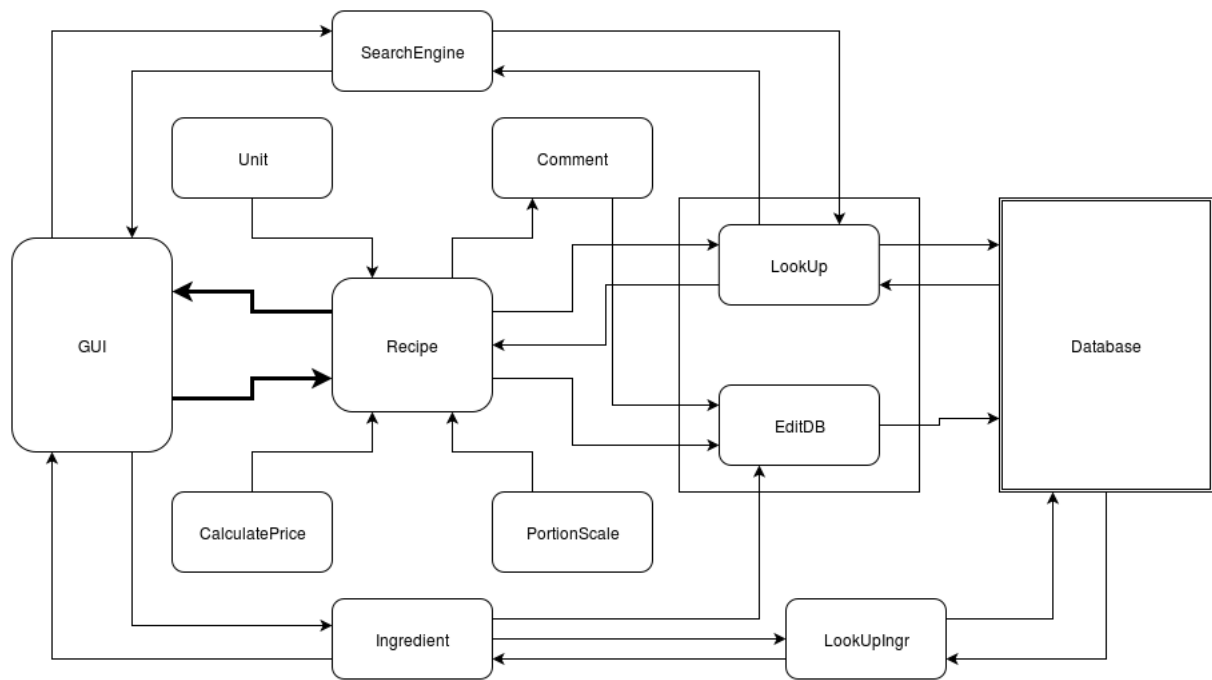
<b>1</b>	<b>Inledning</b>	<b>1</b>
<b>2</b>	<b>Arkitektur</b>	<b>2</b>
2.1	Databas . . . . .	2
2.2	Recipe . . . . .	3
2.3	GUI . . . . .	3
2.4	SearchEngine . . . . .	3
2.5	Unit . . . . .	3
2.6	Comment . . . . .	3
2.7	CalculatePrice . . . . .	3
2.8	PortionScale . . . . .	3
2.9	Ingredient . . . . .	3
2.10	LookUp, LookUpIngr . . . . .	3
2.11	EditDB . . . . .	3
<b>3</b>	<b>Detaljerad teknisk specifikation</b>	<b>4</b>
<b>4</b>	<b>Specifikation av protokoll</b>	<b>5</b>
<b>5</b>	<b>Design av användargränssnitt</b>	<b>6</b>
<b>6</b>	<b>Design av databas</b>	<b>7</b>

# 1. Inledning

Alla har vi någon gång stått framför vårt kylskåp och funderat över vad man kan hitta på för middag med det kylskåpsinnehåll vi konfronteras med. Projektet MatLabb är en interaktiv receptdatabas med grafiskt användargränssnitt. Dess syfte är att hjälpa användaren att organisera recept, söka recept baserat på tillgängliga ingredienser, samt att underlätta portions- och enhetsomvandling. Recepten kommer även att innehålla information om exempelvis näringsinnehåll och pris.

MatLabbs målbild presenteras närmare i dokumentet “MatLabb – Kravspecifikation”, medan detta dokument närmare presenterar skalet och vad som finns under detsamma.

## 2. Arkitektur



Figur 2.1: Översikt över första lagrets moduler.

MatLabb har tre centrala moduler:

1. En databas som lagrar all receptrelaterad information.
2. En receptmodul (**Recipe**) som genom hjälpmoduler hämtar, skriver och behandlar databasens information.
3. Ett användargränssnitt (**GUI**) som hjälper användaren att på ett intuitivt och välbekant sätt interagerar med receptmodulen, och i förlängningen databasen.

I detta kapitel ges en överblick över funktionaliteten hos modulerna i första “lagret” och hur dessa interagerar.

### 2.1 Databas

Databasen utgörs av en MySQL-databas där recept, ingredienser och relaterad data lagras. Exakt data som lagras framgår i kravspecifikationen, samt i kapitel 3 - *Detaljerad teknisk specifikation*.

## 2.2 Recipe

## 2.3 GUI

## 2.4 SearchEngine

## 2.5 Unit

`Unit` är en modul vars syfte är att konvertera enheter. Den ska kunna hantera prefixbaserade enheter (ex. deciliter  $\leftrightarrow$  liter) och bör kunna hantera “köksmått” (ex. matsked  $\leftrightarrow$  tesked). Modulen ska inte kunna hantera omvandling mellan volymenheter och viktenheter då detta förutsätter känd densitet för ingredienserna i fråga, något som sällan är tillgängligt och av föga intresse.

`Unit` används utav `Recipe` i samspel med `PortionScale`, samt i samspel med `CalculatePrice`.

## 2.6 Comment

`Comment` är en modul vars syfte är att hantera receptkommentarer. Modulen används utav `Recipe` för att vidarebefodra relevant information till `EditDB`.

## 2.7 CalculatePrice

`CalculatePrice` är en modul som med hjälp av prisdata

## 2.8 PortionScale

## 2.9 Ingredient

## 2.10 LookUp, LookUpIngr

## 2.11 EditDB

### **3. Detaljerad teknisk specifikation**

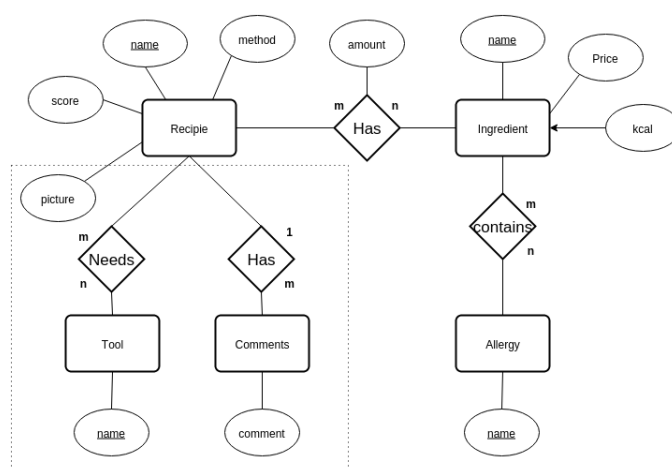
## 4. Specifikation av protokoll



## 5. Design av användargränssnitt

## 6. Design av databas

Då all receptinformation behöver sparas mellan körningar av programmet behöver den lagras externt. I det här programmet kommer en databas användas med hjälp av MySQL. Databasen kan överskådas i EER-diagrammet i 6.1. Den del som ligger inom de streckade området hör till funktionalitet som endast kommer implementeras i mån av tid.



Figur 6.1: Entity-Relationship diagram

Databasen kommer bestå av fem entiteter **Recipe** som innehåller information unik för ett enskilt recept. **Ingredient** som är en lista över de olika ingredienser som databasen innehåller, **Allergy** som är en lista över allergier, **Tool** som är en lista över verktyg samt **Comment** som är en lista över kommentarer till varje recept.

Entiteten **recipe** är den entiteten som lagrar namn, beskrivning, bild, tillagningsmetod, då recept skall ha unika namn för att särskilja dem åt är det receptets namn som agerar primärnyckel.

**recipe** har en m-n relation till **ingredient** vilket ger oss en lista på ingredienser till varje recept. Genom att ha attributen *kcal* och *price* på entiteten **ingredients** istället för **recipe** behöver unik information om portionspris och näringsinnehåll till varje recept inte sparas utan kan räknas ut beroende på vilka ingredienser som ingår. Genom att tillföra attributet *amount* behöver varje ingrediens endast lagras en gång per recept och enhetsomvandling och portionsskalning kommer vara möjlig.

För att hålla ordning på de vanligaste matallergierna (samt kött, mejeri och fisk för veganer/vegetarer) finns entiteten **Allergy**. Den har även en 1-n relation till **Comment** vilket resulterar i att alla recept får en lista med kommentarer skrivna av användaren, samt en m-n relation till **Tool** som ger en lista över de redskap som behövs.

Genom att utforma databasen enligt sagda modell kommer programmet kunna utföra sökningar, olika sökningar och filtreringar på ingredienser som ingår, inte ingår eventuella allergener etc.