

MedImager - Projektplanering

Daniel Ahlqvist, Carl-Henrik Braw, Marcus Linderholm,
Michel Folkemark, Renée Gyllensvaan

Januari/februari 2017

1 Bakgrund

MedView är ett projekt mellan institutionen för tillämpad IT vid Chalmers tekniska högskola och Göteborgs universitet, Oral medicin vid Sahlgrenska akademien och Högskolan i Skövde. Projektet har innefattat bland annat att med bild och text dokumentera patienters hälsa vid oral medicin. Med tiden har en databas bestående av cirka 70 000 bilder byggts upp. Något som dock har saknats är verktyg för att på ett effektivt sätt kunna dra nytta av denna databas. Verktyg för att kunna organisera bilder, dela dem med andra samt kunna samarbeta kring dem skulle kunna utöka användningsområdena för databasen ytterligare. Ett sådant verktyg har efterfrågats av tandexperter vid Sahlgrenska akademien och skulle kunna hjälpa till att förbättra utbildningen vid universitetet.

2 Syfte

Syftet med projektet MedImager är att utveckla en webbaserad applikation för studier av kliniska bilder tagna inom tandvården. Applikationen ska underlätta för orala experter att lära sig av tidigare patienters sjukdomar för kommande fall samt kunna användas i utbildningssyfte. Rapporten ska sedan presentera tillvägagångssättet och den slutliga applikationen.

3 Problem- och uppgiftsanalys

Uppgiften är att ta fram en webbapplikation som realiserar syftet med projektet. Applikationen ska gå att nå över hela världen, men för att komma åt informationen i databasen behövs ett användarkonto. Kontot måste vara verifierat av en administratör för MedImager. I applikationen kan användaren exempelvis söka efter orala sjukdomar eller symptom och finna bilder som stämmer in på dessa. Användaren ska även kunna dela egenskapade uppsättningar av bilder med andra tandexperter samt samarbeta kring dessa.

Den slutliga applikationen kan delas upp i två relativt väl avgränsade delar, backend och frontend. Problemställningar och uppgifter kan därmed lämpligen formuleras med avseende på dessa.

3.1 Frontend

Problemställningar för frontend innefattar:

- Kommunikation med backend.
- Visning av bildgalleri och metadata kopplat till bilderna.
- Användarvänligt grafiskt gränssnitt för interaktion.

Den första punkten innefattar att kunna skicka HTTP-förfrågningar till backend och även att kunna ta emot svar från servern. Frontend ska kunna visa svaren från servern i webbläsaren.

Visningen av bilderna är viktig, då detta är det primära syftet med applikationen. Gallerier ska kunna visas med avseende på sökning och filtrering gjord av användaren. Bilderna ska även presenteras i en genomtänkt ordning, baserat på popularitet och relevans för sökningen. Viss metadata från bilderna ska skrivas ut.

Användargränssnittet ska vara designat på ett målgruppsanpassat vis. Målgruppen är tandexperter och -studenter. Upplägget måste vara intuitivt och lätthanterligt utan förvirrande moment. Funktionaliteten i backend ska kunna kontrolleras och interageras med genom gränssnittet.

3.2 Backend

De primära problemställningarna för backend kan delas upp i:

- Vidarebefordring av bilder från databasen till frontend.
- Hantering av användarinformation för användare av applikationen.
- Hantering av delning av bilder inklusive möjlighet att samarbeta kring dessa mellan flera personer.

Ett rimligt första steg i att vidarebefordra bilder från databasen till frontend är att helt enkelt implementera sökning och hämtning av en godtycklig bild från databasen. Med detta som grund kan man sedan utöka med implementation för att hämta en grupp av bilder samt hämta bilder baserat på parametrar såsom ålder, kön, symptom etc. Nästa steg är att kunna utföra allt detta så optimalt som möjligt, både vad gäller tid och användning av datorkraft. Det sista steget är sedan att paketera de begärda bilderna så att de sedan är redo att skickas till frontend.

Ett första steg i hanteringen av användarinformation är att implementera lagring av användaruppgifter. Detta kommer ske med en relationsdatabas. Därefter måste kopplingen och hanteringen av denna skapas i backendkoden. Detta innefattar bland annat att inmatningar måste valideras och saneras så att de följer bestämda mönster. Exempelvis måste e-postadresser innehålla ett snabel-a och en punkt. Denna databas kommer vara fristående från bilddatabasen, och kommer till skillnad från den, vara både läs- och skrivbar.

Delning av bilder syftar till att användaren ska kunna visa bilder för andra tandexperter eller -studenter. Detta ska kunna ske både genom granskning via MedImager, samt ett enkelt sätt att lägga in bilder i bildskärmspresentationer. Problemställningen här är att implementera sändning och mottagning av bildsamlingar mellan olika användare. Utöver detta ska även flera användare i realtid kunna samarbeta kring en bildsamling. En ytterligare dimension i problemet blir således att korrekt hantera fallet då flera användare samtidigt modifierar viss data.

Gemensamt för ovanstående punkter är att allt måste hanteras på ett säkert sätt. Både när bilder skickas och användarinformation lagras ska detta ske på ett sätt som inte äventyrar datans integritet.

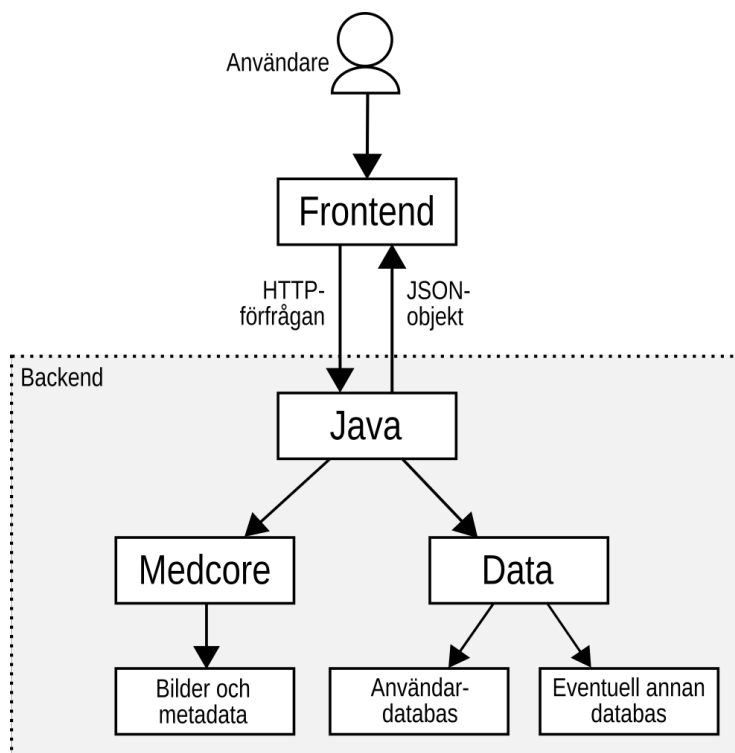
4 Avgränsningar

Den enda riktiga avgränsningen som görs är att använda existerande datastruktur och tillhörande bibliotek istället för att utveckla ett nytt sätt att lagra och interagera med bilddatabasen.

5 Metod/genomförande

Arbetet kommer inledningsvis att delas upp i att en del av gruppen har hand om implementering och design av frontend, medan andra delen av gruppen har hand om implementation av extraheringen av bilder i backend. Detta enligt designmönstret Model-View-Controller.[1]

Flödet för applikationen beskrivs i figur 1.



Figur 1: Diagram över systemet.

5.1 Frontend

Då MedImager kommer vara en webbapplikation förefaller det sig naturligt att frontend designas och byggs upp med hjälp av HTML, CSS och Javascript. Kommunikationen med backend kommer att implementeras med ett REST (Representational State Transfer) API (Application Programming Interface), vilket innebär att gruppen kommer använda Ajax via jQuery för att skicka HTTP(S)-förfrågningar till backend.

Gruppen planerar även att integrera Google Docs, främst presentationsverktyget, i webbapplikationen, eftersom syftet med applikationen är att använda den som ett utbildningsverktyg.

Då inte alla i gruppen är helt bevandrade i verktygen som kommer användas, kommer inledningsvis en del av tiden att ägnas åt inläring. Då alla dessa verktyg är oerhört väldokumenterade på internet bör rätt information relativt enkelt kunna få tag på rätt information.

5.2 Grafiskt gränssnitt

Designen kommer tas fram genom skisser baserade på studier. Bland annat kommer populära webbplatser granskas och litteratur angående grafiska gränssnitt läsas. Även användartest inom målgruppen kommer göras, det vill säga på tan-dexperter, och utifrån dessa kan slutsatser kring förbättringar dras. Själva designen kommer utgå från Googles designstandard, Material Design, men kan komma att modifieras något.

5.3 Backend

Ett existerande bibliotek skrivet i Java finns redan sedan tidigare att tillgå och kommer att agera som grund för applikationens backend. Detta bibliotek har i princip all funktionalitet som kan komma att behövas vid hanteringen av bild-databasen. Det är denna backend som sedan kommer utnyttjas för att extrahera bilder ur bild-databasen på ett sätt som passar applikationens behov. Bilder kommer kunna behöva hämtas baserat på parametrar som exempelvis ålder, kön och symptom. Utöver detta kommer det behöva lagras användarinformation med mera i en databas (lämpligtvis en MySQL-databas).

För kommunikation med frontend används Java Spring, ett serverramverk som idag är marknadsledande[2]. Klientens förfrågningar besvaras med ett REST-

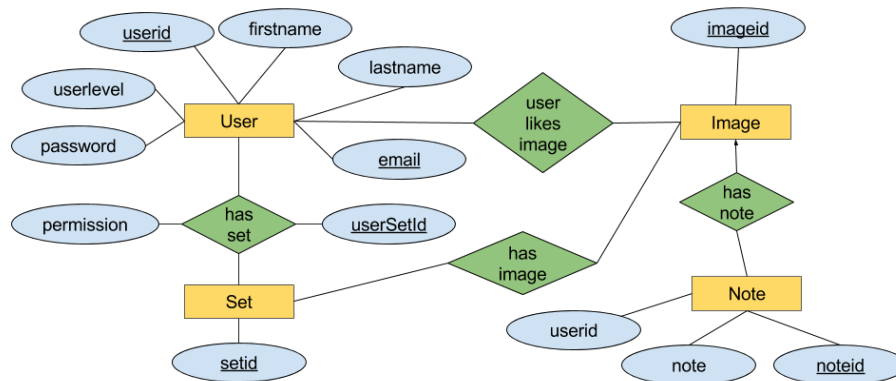
API. Backend kan delas upp i tre komponenter: Spring sköter vår controller, modell i form av en relationsdatabas i MySQL och en view skriven i JSP.

Backend kommer köras på en Ubuntu (16.04 LTS) Virtual Machine, där värden är DigitalOcean, med varierande hårdvara utefter behovet och i vilken fas projektet är i. Beroende på vilket operativsystem som applikationen kommer köras på efter publiceringen kan Docker behöva användas för att säkerställa funktionen.

Inledningsvis kommer större delen av tiden att behöva ägnas åt att bekanta sig med den befintliga backendkoden. En omfattande dokumentation av backend i form av Javadoc finns att tillgå för detta syfte. Även existerande program som med hjälp av denna backend utnyttjar bilddatabasen, såsom programmet MVisualizer, kan komma att vara till stor nytta vid inläring. Källkod till dessa program har gjorts tillgängligt för gruppen. Denna kod bör kunna ge konkreta exempel på hur ett fungerande program hanterar hämtningen av bilder från databasen, vilket troligtvis kan vara en bra startpunkt för inlärningsprocessen.

5.4 Modell

Utöver den databas som innehåller bilderna och dess metadata, behöver webapplikationen en egen databas med information om användare, bildsamlingar och anteckningar till en specifik bild. Relationerna mellan dessa tabeller i databasen beskrivs av entity-relationship-diagrammet (se figur 2).



Figur 2: ER-diagram över databasen.

6 Tidsplan

Det första som kommer göras, utöver att planera projektet, är att börja med skisser av det grafiska gränssnittet, det vill säga designen av applikationen. Kort därefter börjar arbetet med backend och några dagar senare även frontend. Dessa kommer pågå under stora delar av projekttiden. I samband med utvecklandet kommer även rapporten skrivas. I slutfasen av projektet kommer en del tid läggas på att ordentligt testa prestandan hos servern för att den ska kunna hantera en större användarbas. Den föreslagna tidsplanen kan ses nedan i form av ett gantt-schema (figur 3).

Projektindel \ Vecka	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Planering																		
Rapport																		
Oponering																		
Presentation																		
Dokumentation																		
Design																		
Backend																		
Frontend																		
Testning																		

Figur 3: Tidsplan i form av gantt-schema.

Referenser

- [1] Google. Mvc architecture (https://developer.chrome.com/apps/app_frameworks).
- [2] RebelLabs. Java frameworks användningsgrad (<https://zeroturnaround.com/rebellabs/top-4-java-web-frameworks-revealed-real-life-usage-data-of-spring-mvc-vaadin-gwt-and-jsf>).