

# Arkitekturbeskrivning

Redaktör: Jonas Andersson

Datum: 2015-05-06

**Version: 2.2**

## Dokumenthistorik

Datum	Version	Utförda ändringar	Utförda av
2015-02-12	0.1	Dokumentet skapades	Jonas Andersson
2015-03-09	0.2	Utökade arkitekturen kring front-end (avs. 4.3.1) och la till illustration av databasen (avs. 8.1.2).	Jonas Andersson
2015-03-11	0.3	Flyttat ned syftet nedanför innehållsförteckningen, tagit bort indenteringar på underrubriker och lagt till sidnumrering	Pål Kastman
2015-03-25	1.0	Uppdaterat bilderna, lagt till mer information på gör/gör inte, förtydligat hur kartoteket fungerar samt ändrat lite formuleringar.	Jonas Andersson
2015-04-18	2.0	<ul style="list-style-type: none"><li>• Förtydligat beskrivningen av projektet</li><li>• Uppdaterad bilder</li><li>• Förtydligat designfilosofin</li><li>• Tagit bort tidiga prototyper</li><li>• Lagt till bilder från nuvarande system</li><li>• Listat alla paket med kort beskrivning och version</li><li>• Helt skrivit om avs. 6</li><li>• Lagt till lite mer detaljer under avs. 9</li></ul>	Jonas Andersson
2015-05-06	2.1	Korrekturläst och fixat fel stav- och grammatiska fel	Pål Kastman
2015-05-25	2.2	Fixat stav- och grammatiska fel.	Jonas Andersson

# Projektidentitet

Detta dokument gäller för grupp 7 i kursen TDDD77 på Linköpings universitet.

## Projektmedlemmar

Namn	Ansvarsområde	E-post
Daniel Rapp	Teamledare	danth407@student.liu.se
Daniel Falk	Analysansvarig	danfa519@student.liu.se
Jonas Andersson	Arkitekt	jonan111@student.liu.se
Robin Andersson	Utvecklingsledare	roban563@student.liu.se
Albert Karlsson	Kvalitetssamordnare	albka735@student.liu.se
Erik Malmberg	Testledare	erima694@student.liu.se
Pål Kastman	Dokumentansvarig	palka285@student.liu.se

## Kund

Region Östergötland

## Kundkontakt

Daniel Hall, [daniel.hall@regionostergotland.se](mailto:daniel.hall@regionostergotland.se)

Erik Sundvall, [erik.sundvall@regionostergotland.se](mailto:erik.sundvall@regionostergotland.se)

Ingrid Hallander, [ingrid.hallander@regionostergotland.se](mailto:ingrid.hallander@regionostergotland.se)

## Handledare

Lena Buffoni, [lena.buffoni@liu.se](mailto:lena.buffoni@liu.se)

## Examinator och kursansvarig

Kristian Sandahl, [kristian.sandahl@liu.se](mailto:kristian.sandahl@liu.se)

# Innehållsförteckning

[Dokumenthistorik](#)

[Projektidentitet](#)

[Projektmedlemmar](#)

[Kund](#)

[Kundkontakt](#)

[Handledare](#)

[Examinator och kursansvarig](#)

[1. Syfte](#)

[2. Arkitektoniska mål och designfilosofi](#)

[2.1 Översikt av systemet](#)

[2.2 Server & klient](#)

[2.3 Kartoteket](#)

[2.4 Integration med andra system](#)

[2.3 Grundläggande filosofi](#)

[3. Antaganden och beroenden](#)

[4. Signifikanta krav ur arkitektursynpunkt](#)

[5. Beslut, begränsningar och motiveringar](#)

[5.1 Översikt](#)

[5.2 Back-end](#)

[5.3 Front-end](#)

[5.4 Säkerhetskopiering](#)

[5.5 Allmänna anvisningar](#)

[5.5.1 Gör](#)

[5.5.2 Gör inte](#)

[6. Arkitektoniska mekanismer](#)

[6.1 Översikt](#)

[6.2 Handbok](#)

[6.2.1 Sökfunktion](#)

[6.2.2 Lista](#)

[6.2.3 Information](#)

[6.2.4 Administrering](#)

[6.3 Operationsförberedelse](#)

[6.3.1 Plocklista](#)

[6.4 Kartotek](#)

[6.4.1 Vanlig användare](#)

[6.4.2 Administratör](#)

[6.5 Användare](#)

[7. Nyckelabstraktioner](#)

[7.1 Front-end](#)

[7.2 Back-end](#)

[7.3 Model-View-Controller](#)

[8. Lager och ramverk för arkitektur](#)

[9. Arkitektonisk vy](#)

[9.1 Logisk vy](#)

[9.1.2 Modeller](#)

[9.1.3 Stilmallar](#)

[9.1.4 HTML-mallar \(templates\)](#)

[9.1.5 Routes](#)

[9.1.6 Mappstruktur](#)

[9.1.7 Nuvarande paket](#)

[10. Referenser](#)

[10.1 Elektroniska källor](#)

# 1. Syfte

Det här dokumentet beskriver hur vi har byggt upp produkten ur ett arkitekturellt perspektiv. Syftet är att ha all information om arkitekturen samlad och uppdaterad i ett och samma dokument.

## 2. Arkitektoniska mål och designfilosofi

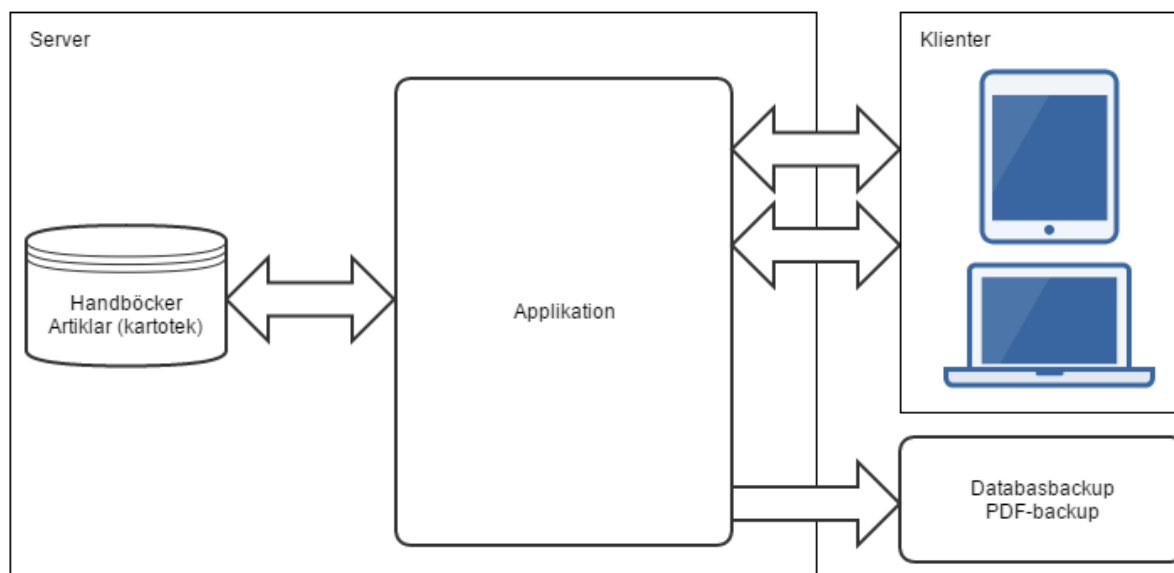
Idag, när Region Östergötland tar hand om en patient i behov av en operation, så använder de sig av pappersbaserade handböcker och checklistor för att förbereda operationen och ta fram rätt operationsutrustning. Detta system ska bytas ut mot en webbaserad applikation som ska kunna köras på både surfplattor och datorer. Det finns även en egenutvecklad artikeldatabas (kartotek) som ska integreras i detta system.

### 2.1 Översikt av systemet

Den webbaserade applikationen innehåller handböcker som beskriver hur man förbereder olika typer av operationer. Handböckerna ska kunna innehålla både text och andra medier som exempelvis bilder, filmer och ljud. Det ska även finnas en lista med artiklar som behövs till operationerna i varje handbok. Artiklarna ska vara kopplade till kartoteket som bland annat innehåller information om var i förråden artiklarna finns placerade.

När en patient registreras så ska en instans av en handbok skapas. I denna instans ska man, från flera enheter samtidigt, kunna checka av en lista med artiklar som ska användas under operationen och även andra förberedelser. En samordnare ska kunna se en översikt på hur långt man har kommit med de förberedelserna för varje instans.

Eftersom det i detta sammanhang finns höga krav på stabilitet så ska applikationen automatiskt säkerhetskopiera handböckerna i ett utskrivbart format.



Figur 1: Grov översikt på systemet

I figur 1 kan man se hur applikationen binder ihop de olika delarna av systemet.

## **2.2 Server & klient**

Applikationen kommer bestå av en server som distribuerar en hemsida till flera klienter. Servern ska kunna köras i en Windowsmiljö. Hemsidan ska vara responsiv och fungera på mobiltelefoner, surfplattor och datorer<sup>1</sup> i olika format.

## **2.3 Kartoteket**

I kartoteket finns information om alla artiklar som Region Östergötland har i förråden. Det finns bland annat information om var artiklarna är placerade samt information relaterade till inköp av artiklar. Det nuvarande systemet ska byggas in i detta system och även vara utbyggbart med lagersaldo i framtiden.

## **2.4 Integration med andra system**

Applikationen ska vara förberedd på att integreras med nuvarande system gällande bland annat inloggning. Därför är det viktigt att denna applikation har en utbytbar/modifierbar användarhantering.

## **2.3 Grundläggande filosofi**

Eftersom kunskaperna bland projektmedlemmarna skiljer sig mycket så är arkitekturen uppbyggd för att få en snabb inlärningskurva. Detta innebär bland annat att mycket färdiga moduler kommer användas för att slippa uppfinna hjulet på nytt. Modulerna kommer bestå av ett enkelt CMS-system tillsammans med små tillägg för att underlätta utvecklingen.

All kod kommer släppas publikt på [github.com](https://github.com)<sup>2</sup>.

## **3. Antaganden och beroenden**

- Tillgång till nuvarande system och exempeldata.
- Tillgång till Kartotekdatabasen.
- Tillgång till hårdvara liknande det kunden har tänkt sig att använda vid eventuell driftsatt system.

## **4. Signifikanta krav ur arkitektursynpunkt**

Se kravspecifikationen för projektet.

---

<sup>1</sup> I detta projekt kommer vi hålla oss till datorer med webbläsaren Internet Explorer samt surfplattor och mobiler med iOS.

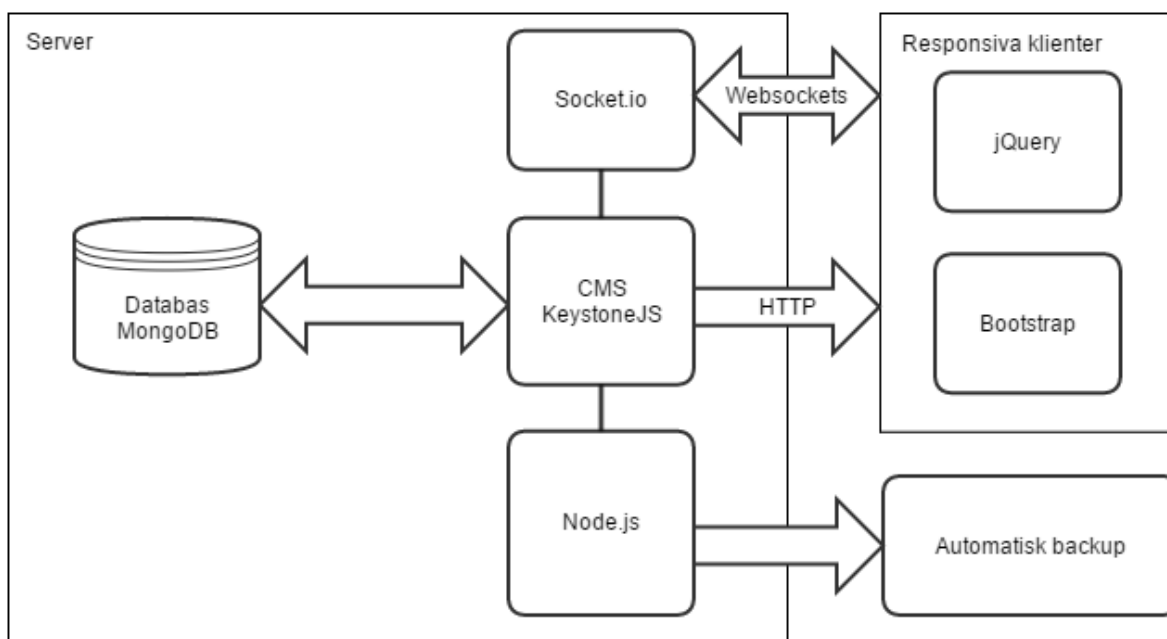
<sup>2</sup> <https://github.com/jcmandersson/Operation> (2015-04-18)

## 5. Beslut, begränsningar och motiveringar

Nedan följer beskrivningar och motiveringar till de tekniker som ska användas till projektet.

### 5.1 Översikt

Programmet är uppdelat i två delar, en serverdel och en klientdel. Serverdelen består av databaskopplingar som kopplas ihop och distribueras ut genom hemsidor till klienterna.



Figur 2: En översikt över arkitekturen.

I figur 2 kan man se en översikt över de mest betydelsefulla tekniker och bibliotek som används för att bygga upp produkten.

### 5.2 Back-end

Koden till servern kommer skrivas till största del i Javascript. Dels för att underlätta inlärningskurvan genom att använda samma språk som på klienten och dels för att realtidskommunikation mellan klienterna underlättas. Grunden till programmet är Node.js<sup>3</sup> vilket är en plattform för att utveckla självständiga program i Javascript med inbyggd pakethanterare. Pakethanteraren gör det enkelt att installera externa bibliotek.

Det största och mest betydelsefulla ramverket för detta projekt är KeystoneJS<sup>4</sup>. Detta är ett CMS-ramverk till Node.js. Följande är några punkter KeystoneJS underlättar:

- Hjälper till att abstrahera systemet. Se MVC under avsnitt 7.3.
- Skapar automatiskt en administreringssida för varje databasmodell. Detta underlättar

<sup>3</sup> <http://nodejs.org/> (2015-04-18)

<sup>4</sup> <http://keystonejs.com/> (2015-04-18)



under utvecklingen då man enkelt kan redigera, ta bort och lägga till data i databasen. Huvuddelen av detta kommer dock att bytas ut eftersom den automatgenererade sidan inte passar till alla ändamål.

- Har ett inbyggt användarsystem som är lätt att modifiera och byta ut.
- Sköter all kommunikation över http-protokollet till klienterna, dvs. gör hemsidan åtkomlig.
- Har inbyggt stöd för templatespråk som exempelvis Handlebars<sup>5</sup> och Less<sup>6</sup>.

För realtidskommunikation kommer ett programmeringsinterface vid namn socket.io<sup>7</sup> användas. Det är väldigt smidigt eftersom det väljer automatiskt hur datan ska skickas beroende på vilken webbläsare som används och vad den stödjer. Socket.io är event-baserat vilket betyder att man skapar events på antingen klient eller serversida som man sedan kan trigga från motsatt sida. Vanliga Javascript-object kan skickas tillsammans med eventen.

## 5.3 Front-end

På klientsidan används bootstrap<sup>8</sup> och jQuery<sup>9</sup>. Bootstrap har många färdiga CSS-klasser så man behöver inte skriva lika mycket CSS själv. De klasser som finns är lätta att använda för att skapa responsiva hemsidor. All kod är dessutom testad för att fungera på olika webbläsare vilket kan vara krångligt att lösa om man skriver all CSS från grunden. JQuery används för att lättare hämta ut ett element på hemsidan och ändra data i det. Det finns även många bra jQuery-bibliotek att hämta som gör att man slipper "uppfinna hjulet" i många fall.

För att ytterligare underlätta design av hemsidan kommer Less att användas. Det är en påbyggnad till CSS som kompileras till vanliga CSS-filer. Fördelen med Less är bland annat att man kan använda variabler och enkla funktioner. Exempelvis kan variablerna användas för att spara de olika färgerna på hemsidan för att enkelt kunna byta ut dem.

### 5.3.1 Struktur

En nackdel med jQuery gentemot andra bibliotek som exempelvis angular<sup>10</sup> är att det lätt blir ostrukturerad kod. För att strukturera så bra som möjligt är det viktigt att koden uppdelad. För att uppnå detta så kommer varje enskild HTML-fil att ha minst en skriptfil och en CSS-fil. På vissa sidor är det lätt att det blir stora skript och då bör man dela upp skriptfilen. Den bästa lösningen är att abstrahera delar av skriptet till egna biblioteksfiler som man kan ladda in och använda.

För att skriptfilerna enbart ska ladda en enskild sida ska filnamnen läggas i en lista i variabeln "locals.scripts". CSS-filerna ska på samma sätt länkas i variabeln "locals.css".

---

<sup>5</sup> <http://handlebarsjs.com/> (2015-04-18)

<sup>6</sup> <http://lesscss.org/> (2015-04-18)

<sup>7</sup> <http://socket.io/> (2015-04-18)

<sup>8</sup> <http://getbootstrap.com/> (2015-04-18)

<sup>9</sup> <http://jquery.com/> (2015-04-18)

<sup>10</sup> <https://angularjs.org/> (2015-04-18)

```
locals.scripts = [  
  'script1.js',  
  'script2.js'  
];  
  
locals.css = [  
  'style1.css',  
  'style2.css'  
];
```

*Kodexempel 1: Exemplet visar hur man länkar skript och CSS-filer.*

## 5.4 Säkerhetskopiering

Ett system för automatisk säkerhetskopiering i form av en pdf-fil för varje handbok kommer att finnas. Varje gång en handbok ändras kommer pdf-filen att uppdateras för att matcha handboken. Pdf-filen ska vara utformad så att operationerna kan förberedas utan hjälp från hemsidan.

## 5.5 Allmänna anvisningar

Nedan följer två listor över vad vi har fokuserat på när vi utvecklat denna produkt.

### 5.5.1 Gör

- För att produkten ska kunna släppas som öppen källkod, så måste alla externa bibliotek enbart bestå av öppen källkod. Kolla därför över licensen på alla externa bibliotek som används.
- För att göra det enkelt att sätta sig in i projektet och byta ut enbart vissa delar, så är det viktigt att man abstraherar så mycket som möjligt. På så sätt kan kunden byta ut delar som inte passade dennes behov.
- Följ designmönstret MVC (beskrivet i avsnitt 8). Av samma anledning som ovan så är det viktigt att följa samma mönster i sin programmering. Det är annars svårt att ta över utvecklingen av projektet.

### 5.5.2 Gör inte

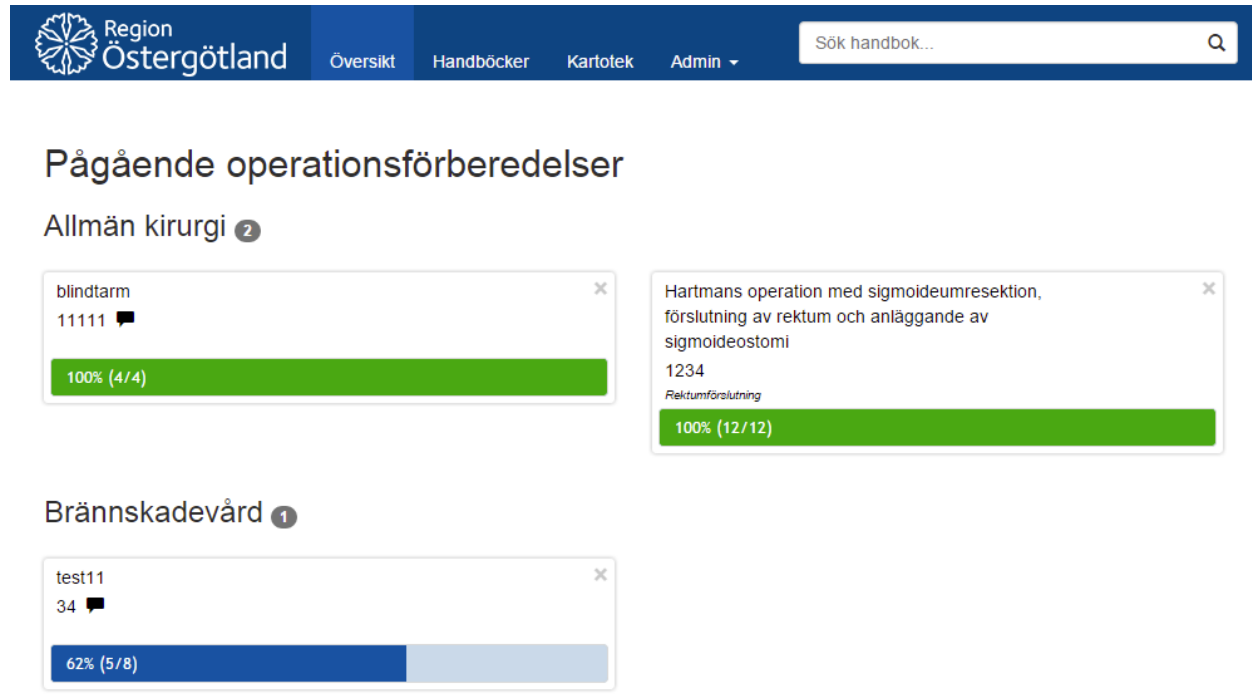
- Använd inte onödigt stora paket för små uppgifter. Detta för att inte göra systemet onödigt komplicerat. Ibland beror även paketen på varandra och det kan bli svårt att uppdatera ett enskilt paket vid behov. Det kan bland annat leda till konstiga felmeddelanden som tar lång tid att lösa.

## 6. Arkitektoniska mekanismer

Här beskrivs syftet och funktionalitet hos olika delar av arkitekturen.

## 6.1 Översikt

Som tidigare nämnts finns en översikt över hur alla operationsförberedelser. Denna sida visar alla operationsförberedelser och hur långt är fortskridna. Här används socket.io för att hela tiden hålla information uppdaterad.



Figur 5: Bild på översiktsvyn

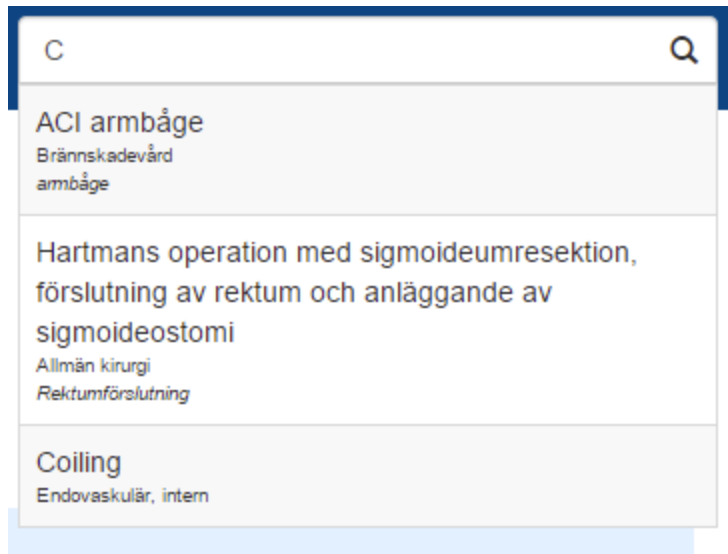
I figur 5 kan man se hur förberedelserna är kategoriserade beroende på vilken kirurgisk specialitet handboken tillhör. Ibland är det olika samordnare beroende på specialitet och det är då enkelt för en samordnare att hitta de operationerna som personen är ansvarig över.

## 6.2 Handbok

En handbok innehåller information om en operationsförberedelse.

### 6.2.1 Sökfunktion

Ett krav är att det ska vara lätt att hitta en handbok. Därför kan man söka både på operationens namn men även på alternativa sökord (taggar), dessa är bra att ha då de medicinska termerna ibland kan vara svåra att komma ihåg.



Figur 6: Bild på sökfunktionen

I figur 6 kan man se sökresultaten där namnet på operationen står i större storlek, och sökorden kursivt i mindre storlek.

### 6.2.2 Lista

Applikationen innehåller en enkel lista med alla handböcker. Den går att sortera på valfri kolumn och kan även gruppera beroende på vilken specialitet handboken tillhör.

### 6.2.3 Information

All information i en handbok är uppdelad i olika rubriker. Dessa rubriker kan i sin tur vara uppdelade i olika processer.

## Publicerad Handbok

### Coiling

Specialitet: Endovaskulär, intern

All information

Preoparea

Anestesi

Allmänt

Operation

Plocklista

#### Övrigt ANE



Manövreringsvred

Använd boardets manövreringsvred för att vrida bordet från röntgenutrustningen.

#### Premedicinering

Paracetamol och Oxycontin

#### Blodrekvisition

Blodgruppering och bastest

#### Antibiotika

Enligt ordination.

Figur 7: Bild på handboksvyn

I figur 7 ser man dels de olika processerna (Preoparea, anestesi, allmänt och operation) samt rubrikerna som hör till processen Anestesi (Övrigt, Premedicinering, blodrekvisition och antibiotika). Man kan även se att handböckerna har stöd för bilder.

#### 6.2.4 Administrering

I administreringsvyn kan man utöver att redigera all information även sortera processer och rubriker genom att dra och släppa dem. Redigeringen under rubrikerna använder sig av en wysiwyg-editor<sup>11</sup>. Innan en redigerad eller ny handbok publiceras måste den granskas av en annan person.

<sup>11</sup> wysiwyg-editor ("What you see is what you get") är ett redigeringsverktyg för enkelt kunna lägga in bilder, ändra stil på text osv. utan att behöva skriva någon kod.

## 6.3 Operationsförberedelse

En operationsförberedelse är nästan likadan som en handbok. Det som skiljer är att vissa av rubrikerna kan gå att kryssa av när de är klara.

 Region Östergötland

ÖversiktHandböckerKartotekAdmin

Sök handbok...

☒ **Hartmans operation med sigmoideumresektion, förslutning av rektum och anläggande av sigmoideostomi**

All informationAnstesiPreopareaPlocklista

Figur 8: Bild på förberedelsevyn

I figur 8 kan man se att rubrikerna går att kryssa av när de är klara. Jämför med figur 7 där rubrikerna inte går att kryssa av.

### 6.3.1 Plocklista

En operationsförberedelse har även en plocklista med artiklar som behövs till operationen.

## Operationsförberedelse

### ☒ 4-kärlsangiografi

Specialitet: Endovaskulär, intern

Linda-ID: 0

	Antal	Namn	Förråd	Sektion	Hylla	Fack	
<input checked="" type="checkbox"/>	1	Aci-slang dagset	SS	A	1-2		<input type="button" value=""/>
<input checked="" type="checkbox"/>	1	Artärkanyl 45Mm. Steril	NO	50	E		<input type="button" value=""/>
<input checked="" type="checkbox"/>	1	Introducer Peel away 14cm x 0.38mm	SO	34	A	1	<input type="button" value=""/>
<input type="checkbox"/>	1	Cvk-Set Venkateter 1.2X 90 Mm S Selacon-T Med Integrerad Flowswitch Grön	NO	15	G	1	<input type="button" value=""/>
<input type="checkbox"/>	1	SPRUTA 50ML R SPRUTPUMP STERIL	SO	29	G		<input type="button" value=""/>

Figur 9: Bild på plocklista-vyn

I figur 9 kan man även se att det finns stöd för att lägga en kommentar på en artikel. Det är vanligt att en artikel är slut eller utbytt och man kan då lägga en kommentar på varför man inte kunde hämta den artikeln och även hur man har löst det istället.

## 6.4 Kartotek

Kartoteket är en databas med artiklarna som bland annat används till engångsmaterialet.

### 6.4.1 Vanlig användare

De vanliga användarna kan enbart se information som behövs för att hitta artiklar i förråden.

### 6.4.2 Administratör

En administratör kan lägga till, ta bort och redigera artiklar. Administratörer kan även se alla fält i databasen.

Region Östergötland							
Översikt Handböcker Kartotek Admin ▾							
Sök artikel... <input type="text"/>							
☑ Visa alla							
Namn	Förråd	Sektion	Hylla	Fack	Pris	Artikelnummer	Återförsäljare
<input type="text" value="Namn..."/>	<input type="text" value="Förråd..."/>	<input type="text" value="Sektion..."/>	<input type="text" value="Hylla..."/>	<input type="text" value="Fack..."/>	<input type="text" value="Pris..."/>	<input type="text" value="Artikelnummer (s)"/>	<input type="text" value="Återförsäljare.."/>
36 Ligacip extrea B	SS	53	F		9673		LT200
ADU Mellanstycke Gasadapter	SO	46	A	3	2143	28869	
ADU Samplingslang 150cm	SO	46	A	2	9225	29913	M1031274
ADU Samplingslang 300cm	SO	46	B	3	3906	27201	

Figur 10: Bild på administreringsvyn i kartoteket

## 6.5 Användare

Det finns många olika typer av användare på sidan. I nuläget finns endast två roller, vanliga användare och administratörer. En vanlig användare kan bara läsa medan en administratör kan ändra all information på sidan. I framtiden kommer dessa roller delas upp. Det kan exempelvis finnas en roll bara för att granska handböcker innan de publiceras. Som tidigare nämnts finns det samordnare som kollar så förberedelserna utförs som de ska, vilket även det skulle kunna vara en egen roll på hemsidan.

## 7. Nyckelabstraktioner

Här listas viktiga abstraktioner för programmet.

### 7.1 Front-end

Det användarna ser kallar vi front-end.

### 7.2 Back-end

Allt som sker på servern kallar vi back-end. Det är alltså back-end som gör att användaren får upp sidan och som kopplar samman flera användare med varandra. Det är även i back-end som databasen ligger. All information som flera användare delar sparas i databasen.

### 7.3 Model-View-Controller

Model-View-Controller, förkortat MVC, är ett designmönster för att abstrahera det användaren ser (view), med själva datan (model) och själva logiken i programmet (controller). Om man använder MVC fullt ut är det lätt att byta ut en av komponenterna, exempelvis byta design (view).



## 8. Lager och ramverk för arkitektur

Som tidigare beskrivet följer projektet designmönstret Model-View-Controller<sup>12</sup>. Model är all data och sköter kopplingen mot databasen. Controller kopplar ihop models och hanterar datan. View är själva grafiska representationen som användaren ser och tar även indata från användaren. Kort beskrivet så hämtar en controller data från en model och hanterar den sedan. När kontrollern är klar skickar den resultatet till en lämplig view som beskriver datan för användaren.

Det finns ett till lager i form av “back-end” och “front-end”. Front-end är i princip samma sak som en view och back-end är allt som händer men inte syns, i detta fall model och controller. Anledningen till att introducera front-end och back-end begreppen och inte bara använda sig av MVC är att det är väldigt enkelt att skilja på front-end och back-end. Front-end är det som sker på klienten och back-end det som sker på servern.

---

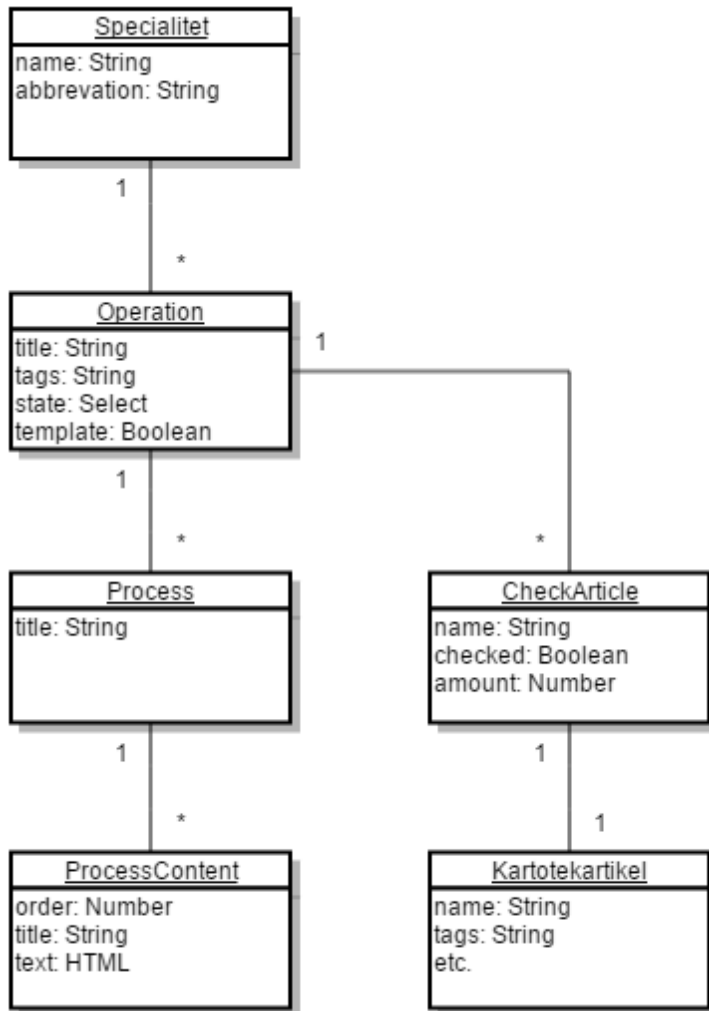
<sup>12</sup> Läs mer på <http://blog.codinghorror.com/understanding-model-view-controller/> (2015-04-18)

## 9. Arkitektonisk vy

Nedan följer beskrivningar av den logiska vyn av arkitekturen.

### 9.1 Databasstruktur

Nedan ser ni en illustration på hur databasen är uppbyggd.



Figur 11: En översikt över databasstrukturen

Som figur 11 visar har varje operation en specialitet. Varje operation kan ha flera processer som i sin tur kan ha flera olika texter i form av ProcessContent samt checkboxar i form av CheckPrepare. En operation kan även ha flera CheckArticles som fungerar som en koppling mot kartoteket. Om något skulle tas bort i kartoteket så finns grundläggande data kvar i CheckArticle. Grundläggande data kan exempelvis vara namn.

## 9.2 Modeller

Databasmodellerna beskrivs av en varsin Javascript-fil i mappen models. Här beskrivs alla fält som ska finnas i en modell. Man kan även lägga till funktioner som hör till modellerna här. Exempelvis har vi en funktion som utifrån en operationsmodell räknar fram hur långt i förberedelserna man kommit. Den hämtar alltså alla artiklar i plocklistan samt alla rubriker som går att checka av och räknar hur många som är checkade.

Alla modellerna har automatiskt följande fält:

Fält	Beskrivning
_id	Ett slumpvist unikt id.
slug	Ett unikt id som bygger på valfritt fält på modellen. Fördelen med slug gentemot _id är att slug är mänskligt läsbart och inte bara slumpvalda tecken som _id.
createdAt	Tidpunkt när objektet skapades.
createdBy	Vilken användare som skapade objektet.
updatedAt	Tidpunkt när objektet uppdaterades.
updatedBy	Vilken användare som senast uppdaterad objektet.

## 9.3 Stilmallar

Stilmallar består av Less-filer. När en klient behöver en CSS-fil kompileras motsvarande Less-fil till CSS och skickas till klienten. En stor fördel med Less är att man kan skapa variabler så att man exempelvis enkelt kan ändra färgschema, höjd på menyn eller liknande. Alla variabler som är relevanta för flera sidor ligger i `"/public/styles/site/variables.less"`.

I filen `"/public/styles/site.less"` finns all Less-kod som är gemensam bland alla sidor. Sedan har även varje sida en egen Less-fil som ligger i mappen `"/public/styles/site/"`.

## 9.4 HTML-mallar (templates)

Mallar är till för att enkelt lägga till data från databasen direkt ut i en HTML-fil som hämtas av klienterna. En mall beskriver hur en sida ser ut strukturmässigt.

Det finns 3 olika typer av mallar: layoutmall, partiell och sidmall. Layout- och partiellmallar kan återanvändas på flera olika sidor medan en sidmall är unik för en viss typ av sida. En sidmall förlänger alltid en layoutmall.

```
<!DOCTYPE html>
<html>
```

```

<head>
  {{#each css}} <!-- Laddar in alla CSS-filer från en lista i routes-->
    <link href="/styles/{{this}}" rel="stylesheet">
  {{/each}}
</head>
<body>
  <header>{{> meny}}</header> <!-- Laddar in en partiellmall med namn meny-->
  <div class="container">
    {{{body}}} <!-- Plats för sidmall-->
  </div>
</body>

```

*Kodexempel 2: Exempel på layoutmall*

```

{{!< default}} <!-- Allt nedanför läggs in på layoutmallen vid namn "default" -->
<form>
  <!-- exempelvis en registreringsida -->
</form>

```

*Kodexempel 3: Exempel på sidmall (partiellmall är likadan förutom att första raden försvinner)*

Eftersom alla sidor i applikationen har till stor del samma information i <head> så används en layoutmall med den taggen. Den layoutmallen innehåller också den meny och logga som används på alla sidor.

Alla mallar ligger i mappen "templates" i projektet. Sidmallar ligger direkt i denna mapp medan layoutmallar och partiellmallar ligger i undermapparna layouts respektive partials.

## 9.5 Routes

En route beskriver vad som behövs för att ladda en specifik sida. Det är här man kopplar ihop stilmallar och HTML-mallar som hör till en viss sida (se kodexempel 1). Varje sida har en egen routefil som även hämtar in den data som behövs från databasen och skickar vidare den till HTML-mallen.

Dessa routefiler ligger i mappen "/routes/views". Det finns även en fil ("/routes/index.js") som beskriver hur url:erna ser ut till varje specifik sida och vilka användare som kommer åt sidorna.

Till sist finns även en fil med enkla skript som många sidor i fliken ("/routes/middleware.js") kör. Här finns exempelvis information om vad som finns i menyn beroende på vem som är inloggad.

## 9.6 Mappstruktur

Programmet delas upp i mappar beroende på vilken del i MVC det tillhör. Nedan följer en lista på var allt är placerat.

- Root
  - Initieringsfiler
  - Configfiler
  - Projektfiler
  - License och readme
- /models (se avs. 9.1.2).
- /lib
  - Kod som alltid körs i bakgrunden och inte bara när en klient är inne på en viss sida. Exempelvis backupskript och socket.io.
- /node\_modules
  - Alla externa bibliotek.
- /public
  - Alla statiska filer som ska kunna nås av klienter genom http-protokollet.
- /public/fonts
  - Fonter som används på hemsidan.
- /public/images
  - Alla bilder som hör till designen på hemsidan.
- /public/js
  - Alla Javascript-filer som körs på klienterna, detta inkluderar externa bibliotek som exempelvis bootstrap och jQuery.
- /public/styles (se avs. 9.1.3).
- /templates (se avs. 9.1.4)
- /routes/views (se avs. 9.1.5).
- /updates/
  - Ibland måste man ändra saker i databasen när man släpper en uppdatering. Detta görs här. Dessa filer körs första gången man startar servern efter att filerna skapats.

## 9.7 Nuvarande paket

Nedan följer listor med alla paket som används för både back-end och front-end.

### 9.7.1 Paket på serversidan

Nedan följer en tabell med alla paket som används på serversidan.

Paket	Beskrivning	Hemsida	Version
Node.js	En plattform för att köra Javascript-baserade servrar.	<a href="https://nodejs.org/">https://nodejs.org/</a>	0.10.22
KeystoneJS	En CMS-plattform till Node.js.	<a href="http://keystonejs.com/">http://keystonejs.com/</a>	0.2.40
Keystone-rest	Bygger automatiskt upp ett REST-api utifrån databasen.	<a href="https://www.npmjs.com/package/keystone-rest">https://www.npmjs.com/package/keystone-rest</a>	0.5.0
Handlebars	En motor för att bygga HTML-mallar.	<a href="http://handlebarsjs.com/">http://handlebarsjs.com/</a>	2.0.0
express-handlebars	För att använda handlebars med keystone.js.	<a href="https://github.com/ericf/express-handlebars">https://github.com/ericf/express-handlebars</a>	1.1.0
MongoDB	En dokumentbaserad databas.	<a href="https://www.mongodb.org/">https://www.mongodb.org/</a>	3.0.2
Mongoose	Ett programmerings API till mongoDB	<a href="http://mongoosejs.com/">http://mongoosejs.com/</a>	3.8.19
Socket.io	Underlättar realtidskommunikation mellan klienter.	<a href="http://socket.io/">http://socket.io/</a>	1.3.4
Underscore	Utökar Javascript med många bra funktioner.	<a href="http://underscorejs.org/">http://underscorejs.org/</a>	1.7.0
wkhtmltopdf	För att rendera en hemsida till en PDF-fil.	<a href="http://wkhtmltopdf.org/">http://wkhtmltopdf.org/</a>	0.1.5
mkdirp	För att skapa mappar.	<a href="https://www.npmjs.com/package/mkdirp">https://www.npmjs.com/package/mkdirp</a>	0.5.0
async	Utökar Javascript med asynkrona funktioner.	<a href="https://www.npmjs.com/package/async">https://www.npmjs.com/package/async</a>	0.9.0
deasync	Används för att synkronisera asynkron kod.	<a href="https://www.npmjs.com/package/deasync">https://www.npmjs.com/package/deasync</a>	0.0.10
dotenv	Används för att ladda plattformsbaserade variabler.	<a href="https://www.npmjs.com/package/dotenv">https://www.npmjs.com/package/dotenv</a>	0.4.0
moment	Ett bibliotek som underlättar tidhantering.	<a href="https://www.npmjs.com/package/moment">https://www.npmjs.com/package/moment</a>	2.8.1

### 9.7.2 Paket på klientsidan

Nedan följer en tabell med alla paket som används på klientsidan

Paket	Beskrivning	Hemsida	Version
Bootstrap	Ett CSS-bibliotek för att underlätta design av responsiva hemsidor.	<a href="http://getbootstrap.com/">http://getbootstrap.com/</a>	3.2.0
jQuery	Ett Javascript bibliotek för att underlätta hantering av bland annat HTML.	<a href="https://jquery.com/">https://jquery.com/</a>	2.1.1
jQuery UI	En utökning av jQuery som har lite mer avancerade funktioner. Används i detta projekt till dra och släpp för att sortera en lista.	<a href="https://jqueryui.com/">https://jqueryui.com/</a>	1.11.3
Handlebars	En motor för att bygga HTML-mallar.	<a href="http://handlebarsjs.com/">http://handlebarsjs.com/</a>	3.0.0
select2	En mer avancerad alternativruta än "select-taggen" som finns i standard HTML.	<a href="https://select2.github.io/">https://select2.github.io/</a>	0.2.9
TinyMCE	En wysiwyg-editor.	<a href="http://www.tinymce.com/">http://www.tinymce.com/</a>	4.1.9
socket.io	Underlättar realtidskommunikation mellan klienter.	<a href="http://socket.io/">http://socket.io/</a>	1.0
Featherlight	För att kunna få upp bilder i ett större format.	<a href="http://noelboss.github.io/featherlight/">http://noelboss.github.io/featherlight/</a>	1.2.3
jquery-tagsinput	En snygg text-input för att lägga till taggar.	<a href="http://xoxco.com/projects/code/tagsinput/">http://xoxco.com/projects/code/tagsinput/</a>	1.3.3
jQuery-zoom	För att zooma in på bilder.	<a href="http://www.jacklmoore.com/zoom/">http://www.jacklmoore.com/zoom/</a>	1.7.14
TableSorter 2.0	För att sortera tabeller	<a href="http://tablesorter.com">http://tablesorter.com</a>	2.0.5b

## 10. Referenser

Nedan följer de referenser som använts i dokumentet.

### 10.1 Elektroniska källor

[1] Template: Architecture Notebook

[http://epf.eclipse.org/wikis/openup/practice.tech.evolutionary\\_arch.base/guidances/templates/architecture\\_notebook\\_BCD3507B.html](http://epf.eclipse.org/wikis/openup/practice.tech.evolutionary_arch.base/guidances/templates/architecture_notebook_BCD3507B.html) (Hämtad 2015-02-10)

[2] Versioner av alla paket i nuvarande produkt

<https://raw.githubusercontent.com/jcmandersson/Operation/master/package.json>

[3] Senaste versionen av produkten <https://github.com/jcmandersson/Operation>