



# **Min hälsoplan**

## **Software Architecture Document (SAD)**



## Revisionshistorik

Datum	Version	Beskrivning	Författare
2014-02-06	PA1	Första utkastet	PL
2014-02-07	PA2	Språkliga korrigeringar efter granskning. Byte av typsnitt från Arial till Times	PL
2014-02-10	A	Beskrivning av webbklient tillagd.	PL

## Innehållsförteckning

1	Inledning .....	5
2	Syfte .....	5
3	Arkitekturella riktlinjer och krav .....	6
3.1	Riktlinjer .....	6
3.2	Övergripande krav .....	6
3.2.1	Integrerad del av Mina vårdkontakter .....	6
3.2.2	Stöd för flera organisationer (multi-tenancy) .....	6
3.2.3	Mobila applikationer .....	7
3.3	Felhantering .....	7
3.4	Icke funktionella .....	7
3.5	Säkerhet .....	7
4	Systemsamverkan .....	8
5	Process-vy .....	9
5.1	Aktörer .....	9
5.2	Användningsfall .....	9
5.2.1	Vårdpersonal .....	9
5.2.2	Patient .....	9
5.2.3	Applikationsadministratör .....	10
5.2.4	Systemadministratör .....	10
5.3	Funktionell översikt .....	10
5.3.1	Huvudflöde .....	10
5.3.2	Planering .....	11
5.3.3	Genomförande .....	11
5.3.4	Uppföljning .....	12
5.4	Entitetsmodell .....	13
6	Logisk-vy .....	14
7	Fysisk-vy .....	16
8	Implementations-vy .....	18
8.1	Webbserver .....	18
8.1.1	Paketstruktur .....	18
8.1.2	Inställningar .....	19
8.2	iOS Applikation .....	21
8.3	Android Applikation .....	22
8.4	Webbklient .....	22
8.5	Källkodshantering .....	23
8.5.1	Riktlinjer för källkod .....	23
8.5.2	Versionsstrategi .....	23
8.5.3	Bygga och köra .....	24
8.5.4	Automatiska tester .....	25

## Definitioner och Referenser

### Definitioner

Term	Beskrivning
REST	Representational state transfer. En arkitekturgren för utformning av lättviktiga API:er. Se även <a href="http://en.wikipedia.org/wiki/Representational_state_transfer">http://en.wikipedia.org/wiki/Representational_state_transfer</a>
API	Applikation Programming Interface. Gränssnitt till en mjukvarutjänst.
APNS	Apple Push Notification Service. Apples tjänst för att skicka push meddelanden till iOS applikationer.
GCM	Google Cloud Messaging. Googles tjänst för att skicka push meddelanden till Android applikationer.
HTTPS	Secure http. Se även <a href="http://en.wikipedia.org/wiki/HTTP_Secure">http://en.wikipedia.org/wiki/HTTP_Secure</a>
iOS	Apples mobila operativsystem. <a href="http://en.wikipedia.org/wiki/iOS">http://en.wikipedia.org/wiki/iOS</a>
Android	Googles mobila operativsystem. <a href="http://en.wikipedia.org/wiki/Android_(operating_system)">http://en.wikipedia.org/wiki/Android_(operating_system)</a>
SSO	Single Sign On. Man loggar in på ett ställe och får då åtkomst till flera relaterade applikationer.
HTML5	Femte generationens HTML med stöd för avancerad klientprogrammering (JavaScript, CSS) och hantering av multimedia.
JavaScript	Dynamiskt programmeringsspråk som stöds av Webbläsare.
CSS	Cascading Style Sheet. Programmeringsspråk för färg och form som stöds av Webbläsare.
MVK	Mina vårdkontakter. Vårdens e-tjänst för invånare och vårdpersonal. Se även <a href="https://minavardkontakter.se">https://minavardkontakter.se</a>
HSA-ID	En identitet för en person, funktion eller enhet i vårdens elektroniska katalog (HSA).
PDL	Patientdatalagen. <a href="http://www.datainspektionen.se/fragor-och-svar/faq-patientdatalagen">http://www.datainspektionen.se/fragor-och-svar/faq-patientdatalagen</a>
HTTP Cookie	Data som skickas från en webbserver för att sparas av webbläsaren. <a href="http://en.wikipedia.org/wiki/HTTP_cookie">http://en.wikipedia.org/wiki/HTTP_cookie</a>
Git	Distribuerat versions och källkodhanteringssystem. <a href="http://en.wikipedia.org/wiki/Git_(software)">http://en.wikipedia.org/wiki/Git_(software)</a>

### Referenser

Dokument	Länk
Min hälsoplan webb	<a href="http://www.minhalsoplan.se">http://www.minhalsoplan.se</a>
Min hälsoplan programvara	<a href="https://github.com/MinHalsoplan/netcare-healthplan">https://github.com/MinHalsoplan/netcare-healthplan</a>
Min hälsoplan demo	<a href="http://demo.minhalsoplan.se/v2">http://demo.minhalsoplan.se/v2</a>
Affero General Public License	<a href="http://www.gnu.org/licenses/agpl-3.0.html">http://www.gnu.org/licenses/agpl-3.0.html</a>
Java Code Conventions	<a href="http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html">http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html</a>
Apple Code Guidelines	<a href="https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/CodingGuidelines/CodingGuidelines.html">https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/CodingGuidelines/CodingGuidelines.html</a>
GitFlow	<a href="http://nvie.com/posts/a-successful-git-branching-model">http://nvie.com/posts/a-successful-git-branching-model</a>



## **1 Inledning**

Min hälsoplan är en e-hälsotjänst i Mina vårdkontakter där vårdpersonal och patienter tillsammans skapar en hälsoplan. Hälsoplanen innehåller en eller flera aktiviteter som patienten får påminnelse att göra, samt kan han eller hon rapportera in resultat från aktiviteten direkt till personalen. Syftet med Min hälsoplan är att stötta både patient och personal under en kort eller lång uppföljning genom att öka tillgänglighet och kvalitet på information samt skapa en ökad kontinuitet i vårdprocessen.

Se även: <http://www.minhalsoplan.se>

## **2 Syfte**

Detta dokument beskriver ingående den kravbild och teknikval som format programvarans arkitektur. Arkitekturens olika aspekter redovisas genom att beskriva olika vyer som process, logisk struktur, fysisk struktur och implementation.

## **3 Arkitekturella riktlinjer och krav**

### **3.1 Riktlinjer**

Initialt så beslutades om ett antal övergripande arkitekturella riktlinjer som projektet sedan har förhållit sig till under genomförandet.

- Applikation skall baseras på öppen källkod och utvecklas som öppen källkod
- För att maximalt värna om källkodens öppenhet ska licensformen vara Affero General Public License (AGPPL)
- Mobila applikationer ska tillämpa en så kallad hybrid arkitektur där man kombinerar en native Android eller iOS applikation med HTML5. Huvuddelen av gränssnittet levereras då som en webbapplikation och är samma oavsett mobil plattform.
- Applikationsservern ska utvecklas på Java plattformen och använda Spring Framework som programmeringsmodell.
- Apache Tomcat används som applikationsserver
- All information lagras i en relationsdatabas som också ska vara löst kopplad till applikationen. Spring Data används för att abstrahera bort ett direkt databasberoende.
- Applikationen ska tillhandahålla REST API:er som ska vara gemensamma för såväl mobila kanalerna som för webbkanalerna.
- Webbapplikationen ska bygga på en modern HTML5 klientarkitektur
- Webbapplikationen kräver Windows IE Explorer 8 eller senare för att fungera
- Webbapplikationen ska baseras på ramverken JQuery, Bootstrap och Highcharts JS
- Samtliga ledtexter ska kunna konfigureras för att enklare möjliggöra en eventuell översättning till andra språk och ett framtida stöd för flerspråkighet

### **3.2 Övergripande krav**

#### **3.2.1 Integrerad del av Mina vårdkontakter**

- Tjänsten ska vara en del av Mina vårdkontakter med gemensamt utseende och inloggning. En så kallad uthoppstjänst.

#### **3.2.2 Stöd för flera organisationer (multi-tenancy)**

- Tjänsten ska stödja så kallad multi-tenancy, dvs. en instans av applikationen ska stödja samtidig användning av flera olika organisationer. En användande organisation är i detta fall en vårdenhet som kan identifieras med ett HSA-ID.
- Det ska finnas stöd för att dela med sig av aktivitetsmallar mellan vårdenheter. Därför ska vårdpersonal kunna tilldelas behörighet att hantera aktivitetsmallar på olika nivåer som:
  - Den egna vårdenheten
  - Det egna landstinget/regionen
  - Globalt/nationellt tillgängliga



### 3.2.3 Mobila applikationer

- Mobila applikationer för iOS, och Android ska finnas för påminnelser och rapportering av aktiviteter. Dessa ska vara publika och åtkomliga över respektive Appstore.
- Påminnelser ska skickas via gängse mobila meddelandesystem, dvs. Apple APNS och Google GCM.
- Mobila applikationer ska vara av så kallad hybrid typ där majoriteten av gränssnittet baserar sig på en gemensam uppsättning Webb sidor och HTML5. Inloggning med autentisering och påminnelser sker med native-funktioner.
- De mobila applikationerna ska baseras på samma server gränssnitt och REST API som övriga applikationer.

### 3.3 Felhantering

- Fel och varningar ska loggas till en särskild applikationsloggfil där det tydligt framgår vilken nivå felet har (FATAL, ERROR, WARNING)
- När ett fel inträffar ska det skickas till en funktionsbrevlåda för supporten och/eller systemadministratörerna.

### 3.4 Icke funktionella

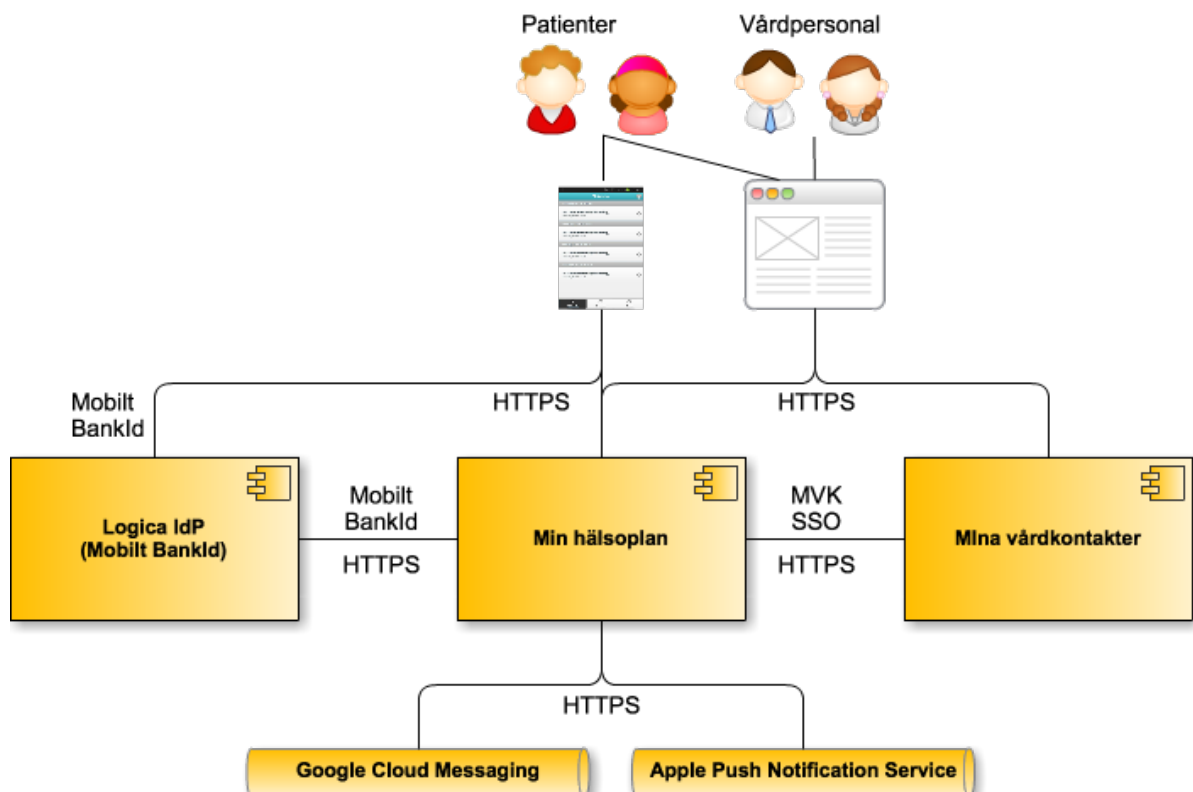
- Sessioner hanteras med HTTP cookies (kakor)
- Det ska vara möjligt att skala applikationen horisontellt genom att lägga till fler noder som lastbalanseras
- Samtliga loggfiler ska roteras (varje dygn)

### 3.5 Säkerhet

- Inkommande och utgående kommunikation sker uteslutande via internet med krypterade förbindelser över HTTPS och port 443 respektive 8890 (Mobilt BankID).
- Autentisering för åtkomst till Webbapplikationen sker via Mina vårdkontakter SSO 2.0 för såväl vårdpersonal och patienter. Det krävs härvidlag autentisering med e-legitimation.
- Autentisering för åtkomst till Mobilapplikation sker med Mobilt BankId.
- Vårdpersonalens åtkomst till patientinformation loggas i enlighet med PDL till en tabell i databasen.
- Extern åtkomst för systemadministration sker med VPN access över SSH.
- Information lagras enbart i den centrala databasen.

## 4 Systemsamverkan

Direkta systemsamband finns med Mina vårdkontakter och Logica IdP (Mobilt BankId). Autentisering för Webbapplikationen sker alltid via Mina vårdkontakter och för Mobilapplikationen via Mobilt BankId. Mobila påminnelser skickas via Google Cloud Messaging (GCM) och Apple Push Notification Service. Samtliga samband körs över HTTPS och förutom Google GCM som använder sig av en API nyckel så krävs det klientcertifikat för åtkomst till övriga system.



*Översikt över direkta systemsamband.*



## **5 Process-vy**

### **5.1 Aktörer**

Huvudaktörerna är:

- Vårdpersonal
- Patient
- Applikationsadministratör
- Systemadministratör

### **5.2 Användningsfall**

#### **5.2.1 Vårdpersonal**

1. Som vårdpersonal vill jag kunna registrera en ny patient med personnummer, namn och telefonnummer
2. Som vårdpersonal vill jag kunna söka efter existerande aktivitetsmallar och då med kriterier som namn, kategori och nivå
3. Som vårdpersonal vill jag kunna skapa nya aktivitetsmallar för min vårdenhet
4. Som vårdpersonal vill jag kunna ge hela regionen/landstinget åtkomst till utvalda aktivitetsmallar på min vårdenhet
5. Som vårdpersonal vill jag kunna ge global/nationell åtkomst till utvalda aktivitetsmallar på min vårdenhet
6. Som vårdpersonal och för en specifik vårdenhet vill jag kunna skapa en ny hälsoplan för min patient
7. Som vårdpersonal vill jag för en hälsoplan kunna planera nya aktiviteter som hämtas från existerande aktivitetsmall
8. Som vårdpersonal vill jag för en hälsoplan kunna redigera aktiviteter som att, exkludera enskilda mätvärden och skattningar, ange tidsperiod och frekvens för aktiviteten samt målvärden
9. Som vårdpersonal vill jag kunna gilla en rapporterad aktivitet och även skicka med en uppmuntran kommentar

#### **5.2.2 Patient**

1. Som patient vill jag kunna rapportera en aktivitet som utförd med mätresultat, skattningar och kommentarer, alternativt som inte utförd
2. Som patient vill jag kunna svara på en gilla/uppmuntran från min vårdkontakt
3. Som patient vill jag kunna följa upp mina resultat som mätvärden och skattningar grafiskt över tiden
4. Som patient vill jag rapportera aktiviteter som utförts extra utöver det planerade schemat
5. Som patient vill jag kunna skriva ut mina resultat
6. Som patient vill jag uppdatera min profil för att korrigera namn, e-post och telefonnummer



7. Som patient vill jag kunna få påminnelser till min mobil om att det är dags att utföra en aktivitet

### 5.2.3 Applikationsadministratör

1. Som applikationsadministratör vill jag kunna lägga upp en ny vårdenhets namn, HSA-ID och landstingstillhörighet
2. Som applikationsadministratör vill jag kunna lägga upp en ny aktivitetskategori för att man senare ska kunna kategorisera och därmed enklare hitta aktivitetsmallar
3. Som applikationsadministratör vill jag kunna lägga till nya måtenheter med namn och lämplig förkortning

### 5.2.4 Systemadministratör

Observera att det inte finns något särskilt gränssnitt eller applikation för att stödja systemadministrationen. Följande användningsfall utförs därför helt manuellt:

1. Som systemadministratör vill jag i databasen kunna sätta behörighetsroll för en viss vårdpersonal för att ge dem möjlighet att skapa aktivitetsmallar på landstings och global/nationell nivå.
2. Som systemadministratör vill jag i databasen kunna tilldela rollen applikationsadministratör till en användare.
3. Som systemadministratör vill jag kunna ta ut PDL-rapporter ur databasen
4. Som systemadministratör vill jag få mail när oväntade applikationsfel uppstår
5. Som systemadministratör vill jag ha åtkomst till servern och loggarna för att kunna felsöka när det uppstår oväntade fel

## 5.3 Funktionell översikt

I dagsläget stödjer e-tjänsten processen från det att en ordination planeras till det att uppföljning av denna plan sker. Det förutsätts att vårdpersonalen och patienten tillsammans planerar hur ordinationen ska genomföras och att det i övervägande andelen fall är patientens första kontakt med tjänsten ifråga.

För att underlätta för vårdpersonalen så finns det funktioner för att skapa aktivitetsmallar som är typiska för enheten, och eftersom tjänsten är tänkt att användas för vitt skilda verksamheter som exempelvis friskvård och dialys så äger varje vårdenhets sina egna uppsättningar med mallar. Mallar kan delas mellan vårdenheter på antingen regional landstingsnivå eller på en nationellt övergripande nivå.

### 5.3.1 Huvudflöde

Vårdgivaren har utfärdat en ordination och träffar patienten, och patienten har fattat ett beslut att använda Min hälsoplan. Det vill säga vårdpersonal och patient kommer överens

om att använda Min hälsoplan som ett stödjande verktyg under genomförandet av ordinationen, som då innebär att följande process stöds:

1. **Planering.** Tillsammans skapar de en plan med lämplig längd och schemalagda aktiviteter
2. **Genomförande.** Under genomförandet rapporterar patienten sina aktiviteter. Vårdpersonalen övervakar att allt går enligt plan och blir varsebliven om konstigheter uppstår som att mätvärden ligger utanför tillåtna intervall eller att rapportering inte sker
3. **Uppföljning.** Efter genomförandet av planen genomförs ett nytt möte som då kan ske med hjälp av videotjänsten och återigen kommer vårdpersonal och patient överens om hur fortsättning ska ske. Som exempel kan man välja att avbryta planen, eller att justera mål och köra vidare under ännu en iteration, eller att ersätta planen med en helt ny

### 5.3.2 Planering

Vårdpersonalen går till Mina vårdkontakter och klickar på länken till Min hälsoplan. Innan planering kan påbörjas måste patienten finnas registrerad på den aktuella vårdenheten, och om inte patienten finns så behöver vårdpersonalen lägga till denne med personnummer, namn och telefonnummer. Nästa steg är att skapa planen som har ett namn och också ett startdatum, varaktighet och de aktiviteter som ska utföras. Förhoppningsvis finns redan en färdig att använda mall till den aktivitet som ska utföras, annars så får vårdpersonalen skapa en ny aktivitetsmall. En aktivitetsmall innehåller vilka mätvärden, frågor, kommentarer och eventuellt om larm ska genereras när ett mätvärde ligger utanför ett angivet intervall, och dessutom om en skattningsskala ska användas vid rapportering för att förmedla någon form av känsla. En plan kan innehålla flera olika aktiviteter. Till aktiviteten kopplas nu målvärden som man är överens om, och dessutom så schemaläggs aktiviteten med veckofrekvens, veckodagar och tidpunkter. När man är nöjd så skapas planen och planen är redo att börja genomföras.

Patienten kan nu med hjälp av sin dator gå till Mina vårdkontakter för att navigera vidare till tjänsten Min hälsoplan. Där finner patienten alla sina planer med ingående aktiviteter, och om patienten vill kan aktivitetsschemat via ett standardformat exporteras till en extern standard kalender funktion. Dessutom kan patienten i sin profil välja att slå på sin mobila kanal och den personliga kod som ska anges för mobil åtkomst. Själva applikationen för den mobila kanalen hämtas via Google Play eller Apple Appstore där man företrädesvis söker på "Min hälsoplan".

### 5.3.3 Genomförande

Under själva genomförandet är fokus framför allt på patienten, dvs. patienten genomför sina aktiviteter och rapporterar sina resultat. Rapporteringen kan både ske via den mobila applikationen och via Mina vårdkontakter. Har patienten valt att använda den mobila kanalen så kommer påminnelser till den mobila enheten ca 15 minuter innan en aktivitet ska genomföras. För varje aktivitet kan patienten se utfallet i grafer där varje mätvärde visas tillsammans med sitt eller sina målvärden från tiden då aktiviteten startade till nu. Det är



dock viktigt att poängtera att tjänsten framför allt handlar om återrapportering och inte stödjer själva genomförandet som sådant. Patienten kan för varje aktivitet välja att exportera alla sina rapporterade värden till sin dator och får då ett format som kan läsas av Excel och liknande program.

Skulle patienten strunta i att rapportera resultat så stängs dessa automatiskt av systemet vilket då personalen på vårdenheten blir varse via ett larm som indikeras på tjänstens förstasida, men patienten har alltid kvar föregående hela veckas aktiviteter (start måndag) och således god tid på sig att rapportera sina aktiviteter innan de stängs.

Under genomförandet är det möjligt för vårdpersonalen att initiera en kort konversation genom att gilla och/eller kommentera en enskild aktivitet, denna kommentar kan då besvaras av patienten ifråga. Syftet med denna konversation är framför allt att från vårdens sida uppmuntra och ytterligare motivera patienterna.

På vårdpersonalens första sida i Min hälsoplan kan man visa rapporteringar för de patienter som har planer på vårdavdelningen. Dessutom listats eventuella larm och kommentarskonversationer.

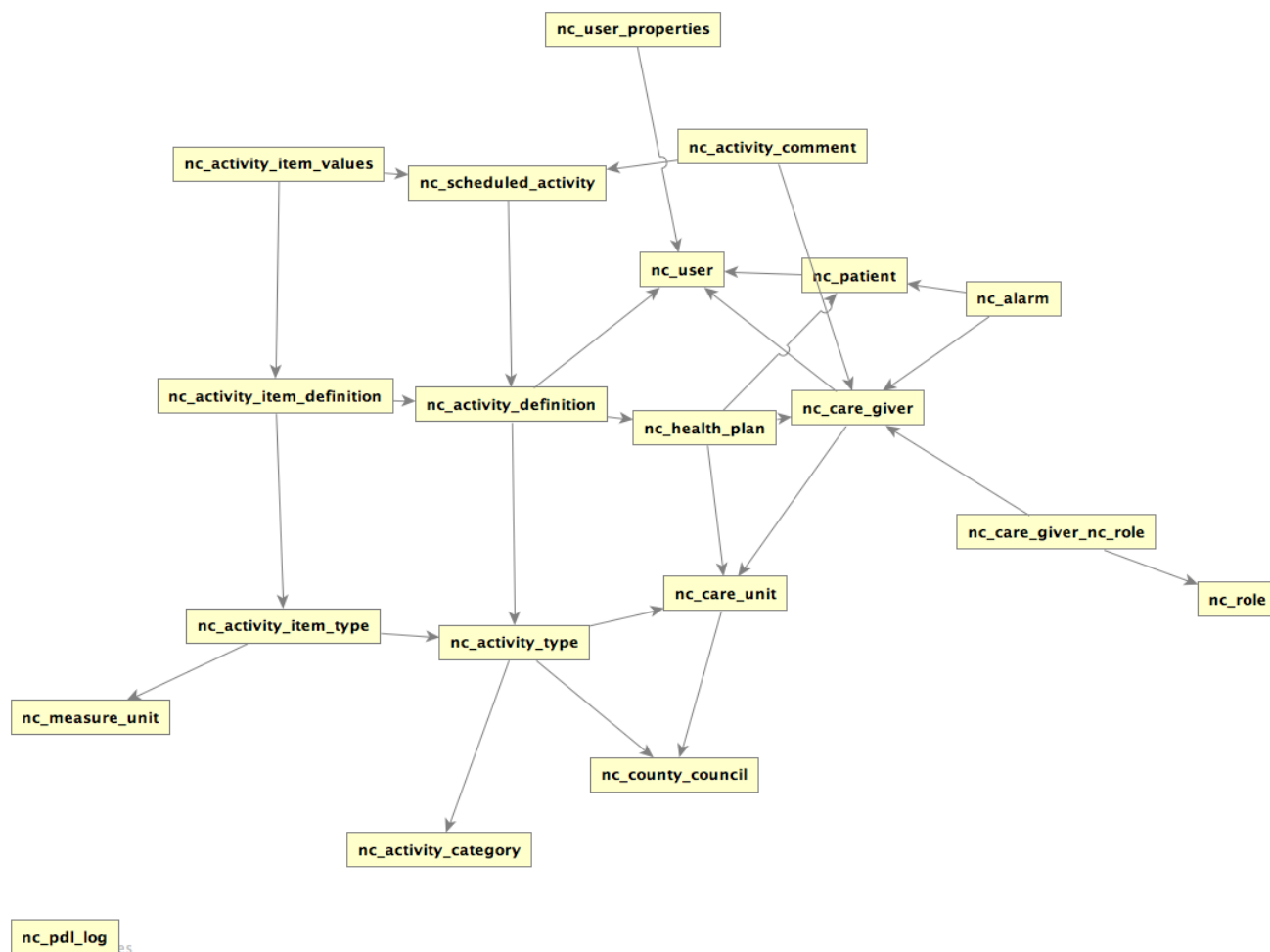
#### **5.3.4 Uppföljning**

Vid en uppföljning som antingen kan ske i ett möte alternativt i ett videomöte går man igenom resultaten tillsammans och vårdpersonalen har via sin sida tillgång till samma resultatinformation som patienten har.

Man kommer överens om åtgärder som att förlänga planen, eller justera målvärden eller helt sonika att avbryta planen för att eventuellt ersätta den med något annat. Det kan också handla om att stänga ned en enskild aktivitet för gott, eller att ersätta en aktivitet med en annan mer ändamålsenlig.

## 5.4 Entitetsmodell

Diagrammet nedan redovisar de entiteter som hanteras i databasen och deras inbördes beroenden.



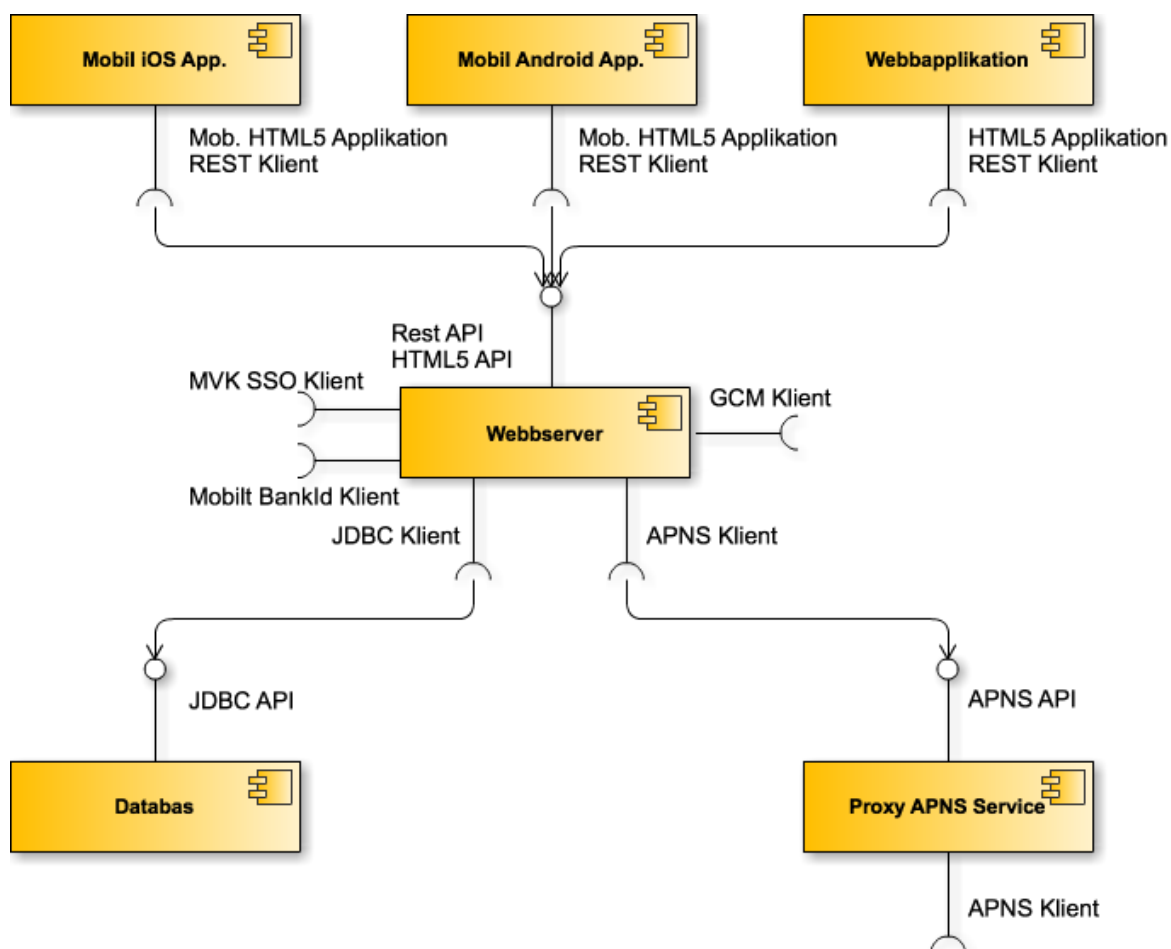
Entiteter som hanteras i databasen med inbördes beroenden. Det kan noteras att PDL loggen ned till vänster inte har några beroenden till andra entiteter. Av läsbarhetsskäl så redovisas inte samtliga egenskaper.

## 6 Logisk-vy

Min hälsoplan består av ett flertal funktionella mjukvarukomponenter. Tekniska ramverk som använts för att realisera dessa funktionella komponenter redovisas inte, men det gör däremot integrationskomponenter och då i form av gränssnitt som tjänstekonsumenter.

Se tabellen och diagrammet nedan för en mer utförligare beskrivning.

<b>Komponent</b>	<b>Beskrivning</b>
Mobil iOS applikation	Native iOS Objective-C applikation som distribueras via Apple Appstore
Mobile Android applikation	Native Android Java applikation som distribueras via Google Play
Mobil HTML5 applikation	Webbapplikation för den mobila kanalen som är en del av Webbservern, se nedan
HTML5 Webbapplikation	Den fullständiga Webbapplikationen för såväl patienter som vårdpersonal
Webbserver	Server applikationen som distribueras som en standard Java Web Application Archive (WAR) och tillhandahåller API:er och HTML5 Webbapplikationer med JavaScript och CSS
GCM Klient	Integrationskomponent för att skicka GCM meddelanden.
APNS Klient	Integrationskomponent för att skicka APNS meddelanden
MVK SSO Klient	Integrationskomponent för att kommunicera med MVK
Mobilt BankId Klient	Integrationskomponent för att kommunicera med Mobilt BankId
Proxy APNS Service	Tillfällig lösning då Java 1.7 inte är kompatibelt med Apples tjänst med avseende på HTTPS. Därför används en intern proxytjänst som kan köras på Java 1.6 plattformen.
Databas	Datalager för all information som applikationen använder, dvs. förutom statiska tekniska konfigurationsfiler.

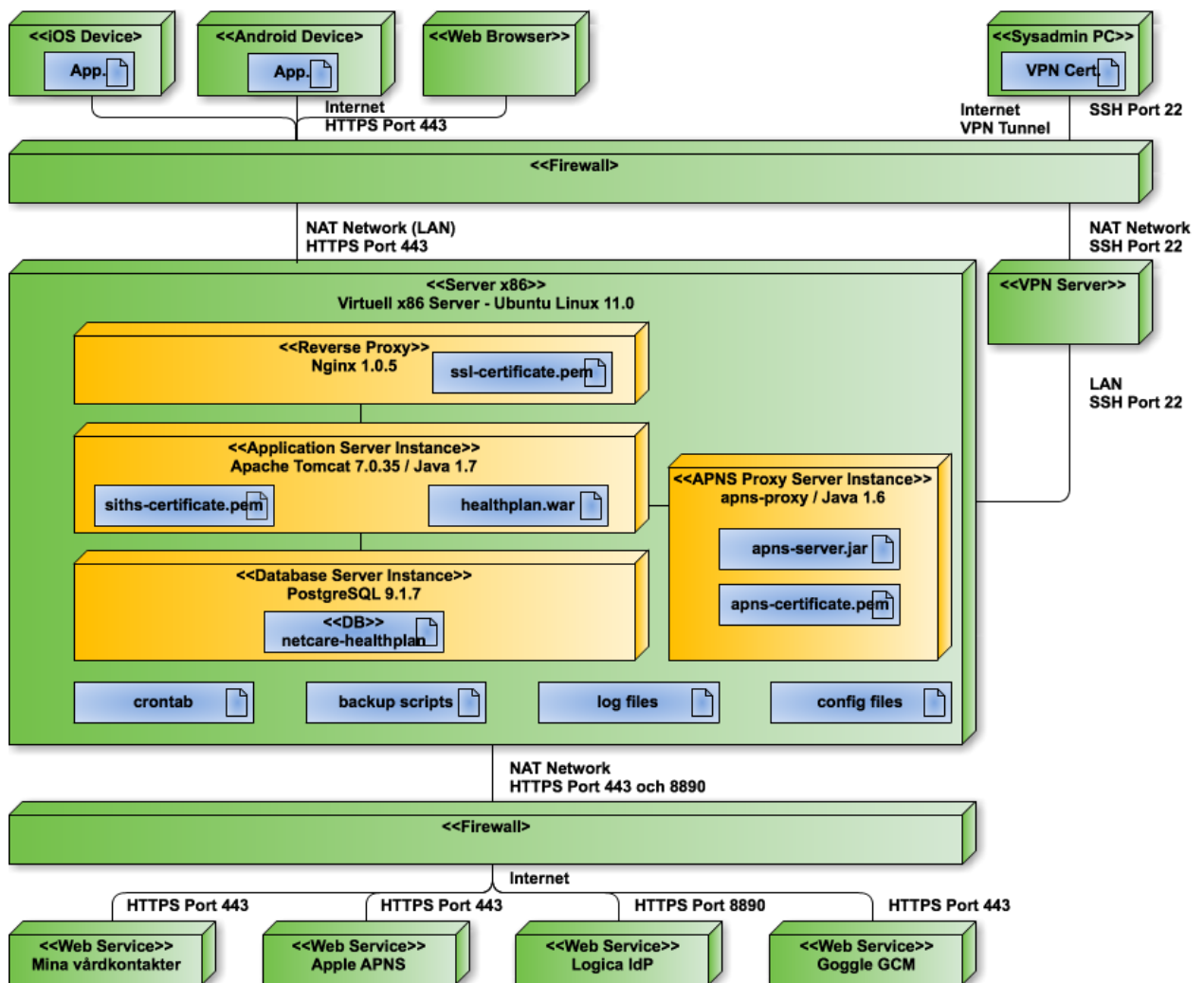


*Logisk vy över involverade mjukvarukomponenter och dess integrationer.*

## 7 Fysisk-vy

Den fysiska vyn kan självfallet variera beroende på krav på tillgänglighet, prestanda och säkerhet osv. Därför beskrivs enbart den verifierade single-server konfigurationen som hitintills har tillämpats.

En typisk server är virtuell och Intel baserad (x86) med 4GB RAM, 4x CPU och 100 GB HDD.



*Fysisk vy med processer och artefakter. För att göra det mer överskådligt så skiljer diagrammet på uppströms/inkommande trafik och nedströms/utgående trafik, men i praktiken kan det vara samma brandvägg för båda riktningarna.*

Mjukvaruplattformen som verifierats är i dagsläget:

- Server
  - OS - Ubuntu Linux 11.0
  - Databas - PostgreSQL 9.1.7





- Java - Java 1.7 (och 1.6 för Proxy APNS)
  - Reverse Proxy - Nginx 1.0.5
  - Applikationsserver - Apache Tomcat 7.0.35
  - Applikation (Min hälsoplan)
- Klient
  - iOS Mobil applikation - iOS 5 och 6, iPhone 4 och 5
  - Android applikation - Android SDK version 2.2 (8) till 4.2 (17)
  - HTML5 applikation – Windows IE version 8 eller senare, Safari, Firefox, Google Chrome



## 8 Implementations-vy

Källkoden kan delas in i 3 olika huvudpaket som i sin tur har beroende till såväl interna och som externa tredjepartskomponenter.

1. Webbserver
2. iOS Applikation
3. Android Applikation

Gemensam för samtliga paket är att de är baserade på öppen källkod, och ingången till samtliga är återfinns i samma källkodsprojekt på GitHub, se <https://github.com/MinHalsoplan/netcare-healthplan>

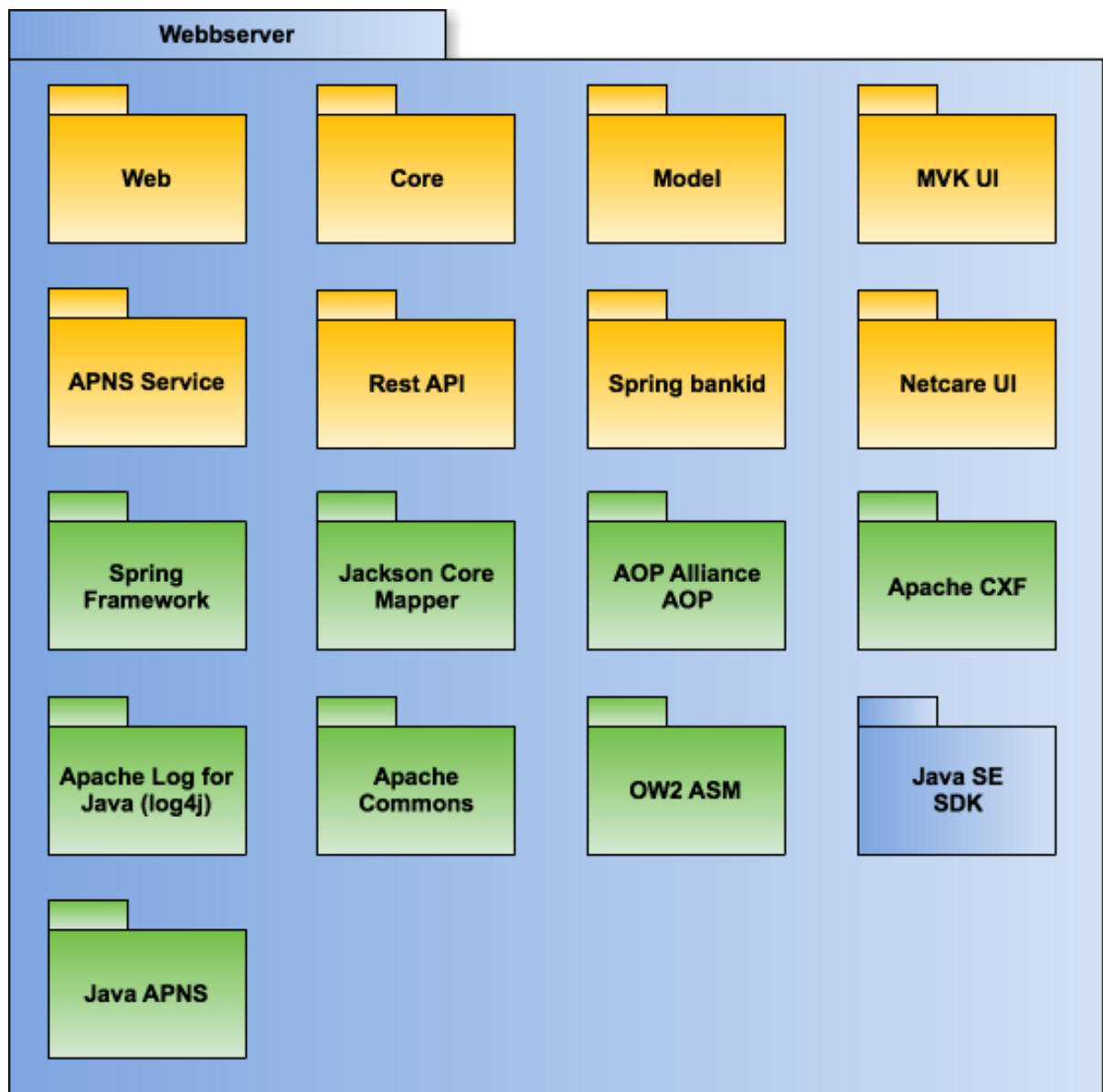
I diagrammen nedan indikerar gula paket källkod som kan utvecklas specifikt för min hälsoplan, gröna paket indikerar att det rör sig om 3:e partsprodukter och blå paket plattformskomponenter.

### 8.1 Webbserver

#### 8.1.1 Paketstruktur

Webbservern består av ett flertal externa tredjepartskomponenter, men också ett flertal olika källkodsprojekt som alla återfinns på GitHub. För att bygga och köra mjukvaran krävs att man har Git 1.7+, Java SE 1.7, och Apache Maven 3.0 installerat.

- BankId komponent
  - <https://github.com/callista-software/spring-bankid>
- Mina vårdkontakter gränssnitt
  - <https://github.com/callista-software/mvk-ui>
- Mina vårdkontakter SSO integration
  - <https://github.com/callista-software/mvk-integration>
- Netcare generiskt gränssnitt
  - <https://github.com/callista-software/netcare-ui>



*Paket som ingår i webbservern. Varje paket manifesteras som en eller flera Java Archive (JAR) filer.*

### 8.1.2 Inställningar

Webbservern har ett antal konfigurationsinställningar som förväntas ligga i användarens hemmabibliotek, dvs. den användare som exekverar applikationen. För en produktionsinstans återfinns inställningarna under biblioteket ~/netcare-healthplan-prod som innehåller följande konfigurationsfiler:

Namn	Beskrivning
application.properties	Applikationsinställningar som funktionsbrevlåda för felmeddelanden och

## Min Hälsoplan - SAD

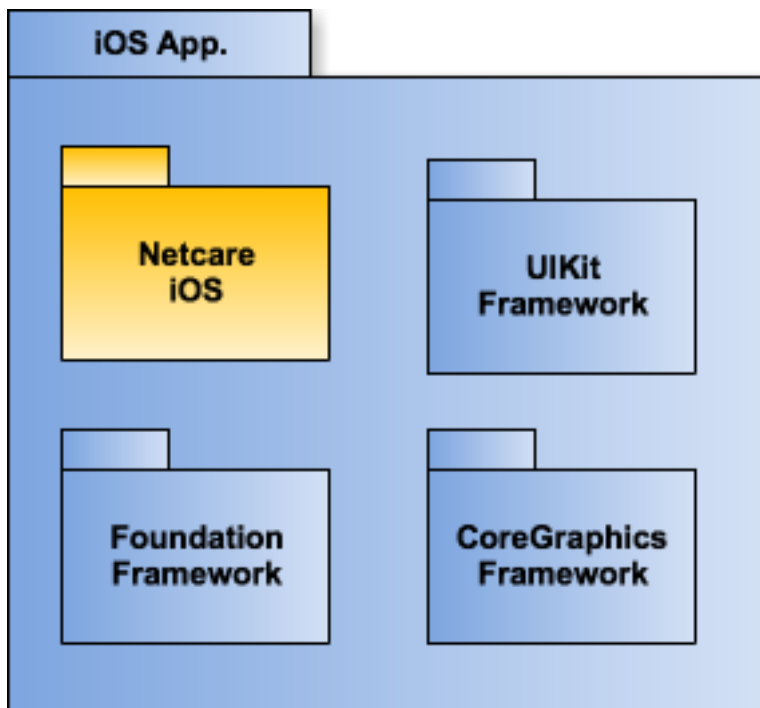
	tid för påminnelser.
bankid.properties	Inställningar för bankid som URL, certifikat och tjänsteid.
messages.properties	Ledtexter på svenska för de texter som kan översättas.
database.properties	Databasinställningar som konto och databasserver.
mvk.properties	Inställningar för MVK som URL, certifikat, och andra parametrar.
push.properties	Inställningar för push notifieringar som API nyckel för GCM och URL och certifikat för APNS tjänsten. <b><i>OBS! Just nu ställs dessa in till att peka på den lokala proxytjänsten, se nedan.</i></b>

Dessutom finns för närvarande inställningar för den temporära Proxy APNS tjänsten och dessa ligger i biblioteket `~/netcare-apns-service`.

<b>Namn</b>	<b>Beskrivning</b>
push.properties	Inställningar för push notifieringar till den riktiga APNS tjänsten som URL och certifikat för APNS tjänsten.

## 8.2 iOS Applikation

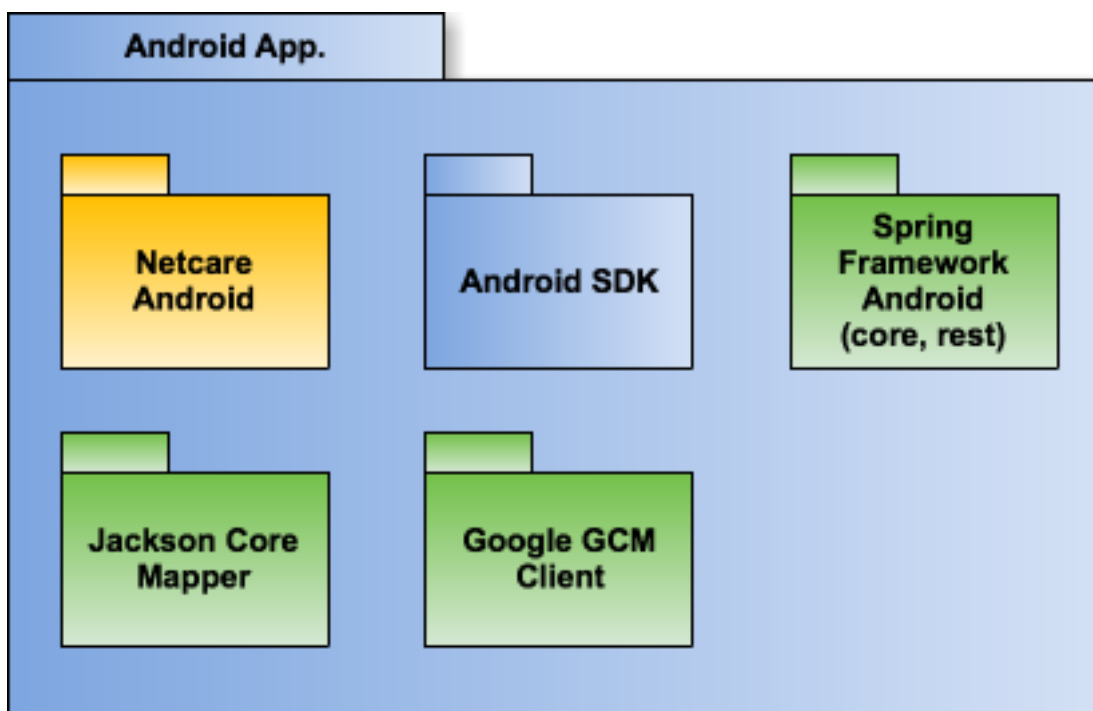
iOS Applikationen är renodlad och har förutom iOS plattformen inga övriga tredjepartsberoenden. För att bygga och utveckla iOS programvaran krävs Apples Xcode och ett utvecklingskonto.



*Paket som ingår i iOS applikationen, endast plattformspaket används.*

### 8.3 Android Applikation

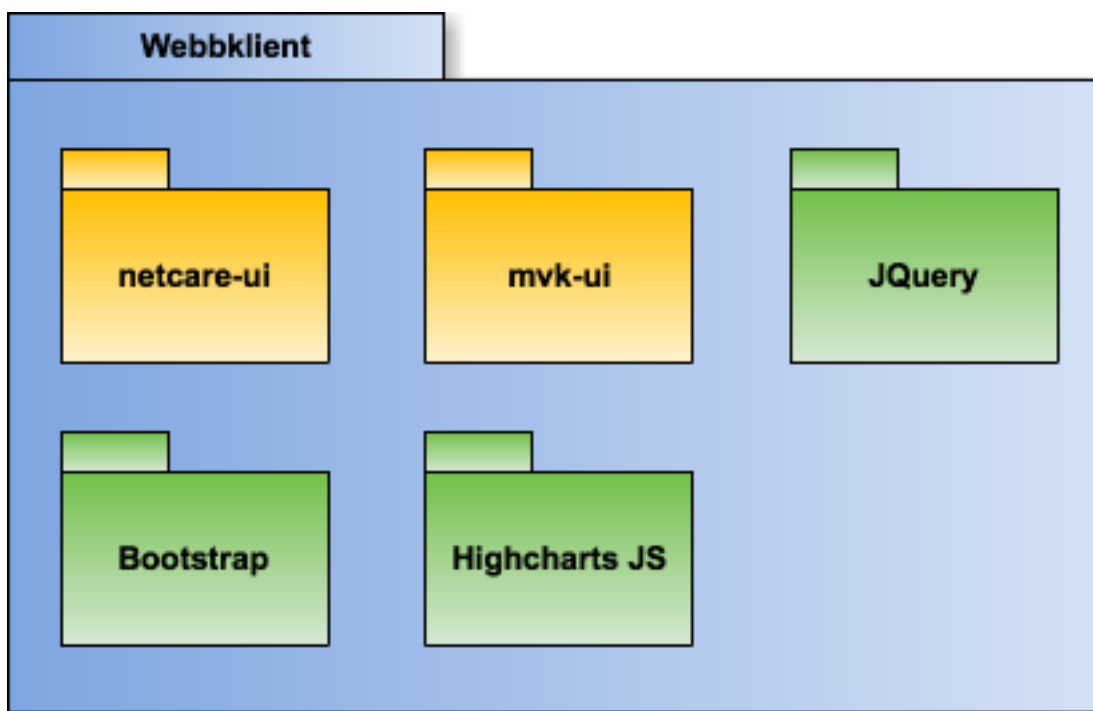
Android applikation har endast några få tredjepartsberoenden. För att bygga och utveckla programvaran krävs att man först installerar Android SDK, Java och Apache Maven.



*Paket som används i Android applikationen.*

### 8.4 Webbclient

Webbclienten körs i en standard läsare som Internet Explorer (IE 8 eller senare), Safari eller Google Chrome. Applikationen är baserad på HTML5 (JavaScript, CSS) där vedertagna tredjeparts öppna JavaScriptramverk används.



*JavaScript ramverk som används för webbklienten.*

Den mobila HTML5 applikationen behöver endast en delmängd av den normala webbklienten.

## 8.5 Källkodshantering

Källkoden är öppen och återfinns på GitHub.

<http://github.com/MinHalsoplan/netcare-healthplan>

Källkoden hanteras i ett Git repository och det är fritt fram för vem som helst att ladda ned och ta del av källkoden.

### 8.5.1 Riktlinjer för källkod

Majoriteten av källkoden är Java och där följs den sedan länge etablerade konventionen som togs fram av SUN Microsystems och nu tillhör Oracle.

<http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html>

För iOS källkod används motsvarande konventioner från Apple.

<https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/CodingGuidelines/CodingGuidelines.html>

### 8.5.2 Versionsstrategi

Under utveckling och underhåll av koden tillämpas en strategi som kallas GitFlow och är en framgångsrik modell när man använder verktyget Git. Strategin beskrivs närmare i detta dokument.

<http://nvie.com/posts/a-successful-git-branching-model>

Kortfattat kan man säga att det alltid finns två källkodsgrenar en för utveckling (dev) och ett huvudspår (master). Det är i dev som all utveckling sker och master är alltid ett snapshot av den senaste releasen och innehåller även så kallade etiketter/taggar för att man ska kunna bevara releasehistoriken. Man hanterar sedan releaser (från dev) och så kallade hot-fixes (från master) i separata kortlivade källkodsgrenar där ändringar för varje ny release förs tillbaka till huvudspåret (master), och eventuella hot-fix ändringar förs tillbaka till utvecklingsspåret (dev).

### 8.5.3 Bygga och köra

För att kunna bygga och köra webbapplikationen krävs det att man har Apache Maven 3.0 och Java SE 1.7 installerad på sin dator.

Kör nedanstående från en lämplig plats på din dator. Notera att det idag finns beroende till en särskild byggservare och därför behövs en del konfiguration i Apache Maven för att processen ska fungera smidigt.

```
$ git clone https://github.com/MinHalsoplan/netcare-healthplan.git
$ cd netcare-healthplan/netcare
$ mvn install
```

Sedan kan man byta bibliotek och köra igång en lokal testinstans av webbapplikationen och det har även genererats testdata i den temporära databasen:

```
$ cd ../netcare-web
$ mvn jetty:run
```

Det kan ta några minuter att starta upp applikationen, och det en hel del loginformation skrivs till terminalen där man startat applikationen. Om allt går bra så ska de sista loggraderna se ut liknande:

```
2014-02-07 10:59:28,707 [...] DEBUG ... - Servlet 'netcare' configured successfully
2014-02-07 10:59:28,719:INFO: ... Started SelectChannelConnector@0.0.0.0:8080
[INFO] Started Jetty Server
```

Om allt ser bra ut så öppna en webbläsare och navigera till

<http://localhost:8080>

Färdigt att använda finns testanvändare och för att logga in som vårdpersonal kan man mata in **hsa-cg-2** eller för att logga in som en patient kan man mara in **191212121212**.

Se även [http://www.minhalsoplan.se/?page\\_id=19](http://www.minhalsoplan.se/?page_id=19) för mer information om testanvändare.





#### 8.5.4 Automatiska tester

Ansatsen är att viktig funktionalitet ska verifieras med automatiska testfall som också körs vid varje bygge. I dagsläget är det företrädesvis serverprogramvaran som verifieras med automatiska tester. Klientapplikationernas gränssnitt verifieras fortfarande manuellt. Det rekommenderas att en byggserver används för att kontinuerligt och vid varje förändring snabbt få feedback på eventuella felaktigheter i källkod och byggsystem.

Testerna utökas allteftersom ändringar och tillägg sker, och för att få en sammanställning över nuläget kan man köra följande kommando från underbiblioteket `netcare`:

```
$ cd ~/netcare-healthplan/netcare  
$ mvn -Daggregate=true surefire-report:report
```

Testrapporten ligger som ett html dokument och öppnas lämpligen i en webbläsare. Se filen `target/site/surefire-report.html` för rapporten ifråga.