

Beteckning: _____



Akademien för teknik och miljö

Utveckling av system för insamling och presentation av tryckskadeförekomster inom sjukvården

Ronnie Lindeberg
Juni 2012

Examensarbetet, 15 hp, B
Datavetenskap

Dataingenjörsprogrammet
Examinator: Bengt Östberg
Handledare: Anders Jackson

Utveckling av system för insamling och presentation av tryckskadeförekomster inom sjukvården

av

Ronnie Lindeberg

Akademin för teknik och miljö
Högskolan i Gävle

S-801 76 Gävle, Sweden

Email:

ndi08rlg@student.hig.se

Abstrakt

På sjukhuset i Gävle utförs regelbundet punktprevalensmätningar på uppkomsten av trycksår. Syftet med dessa mätningar är att hitta effektiva förebyggande åtgärder som kan användas för att minska uppkomsten av trycksår. Mätningarna utförs med hjälp av en surfplatta. För att underlätta insamlingen och presentationen av mätningarnas resultat har det utvecklats ett nytt datasystem. Datasystemet består av en databas med tillhörande databasapplikation. Applikationen är programmerad i Java och databashanteraren är ObjectDB. Datasystemet kan synkronisera och överföra data från surfplattans databas med en USB-kabel. Applikationen har ett användarvänligt grafiskt gränssnitt som ger en arbetsvänlig och enkel miljö att jobba i. Databasens innehåll kan presenteras på tre olika sätt, i en översiktstabell, i olika detaljtabeller och som ett linjediagram.

Nyckelord: PPM, Java, ObjectDB, JPA, statistik

Förord

Den här rapporten beskriver det examensarbete på Högskolan i Gävle som avslutar mina studier på dataingenjörsprogrammet.

Jag vill passa på att tacka Magnus och min handledare Anders Jackson som kom med projektförslaget till examensarbetet och för all hjälp med att svara på de frågor och funderingar jag haft under arbetets gång.

Jag vill också tacka Eva på KTC, Gävle sjukhus.

Avslutningsvis vill jag förtydliga att samtliga mätningars innehåll och siffror som förekommer i rapportens figurer nr 6, 8, 10 och 13 är inte riktiga, utan helt påhittade av mig. Detta på grund av sekretess.

Gävle den 1 juni, 2012

Ronnie Lindeberg,
ndi08rlg@student.hig.se

Innehåll

1 Inledning.....	1
1.1 Bakgrund.....	1
1.1.1 Teoretisk bakgrund.....	1
1.1.2 Allmän bakgrund.....	1
1.2 Problembeskrivning.....	3
1.3 Avgränsning.....	3
1.4 Syfte.....	3
1.5 Frågeställningar.....	3
2 Metod.....	4
2.1 Verksamhetsanalys.....	4
2.2 Problemanalys.....	5
2.3 Funktionsanalys.....	5
2.4 Lösningförslag.....	6
3 Genomförande.....	6
3.1 Planering.....	7
3.2 Synkronisera och överföra mätningar.....	7
3.2.1 Surfplattans databas.....	8
3.2.2 Datasystemets databas.....	8
3.2.3 Lagra data i ObjectDB.....	8
3.2.4 Importera mätningar från surfplattan.....	8
3.3 Presentation av mätningar.....	9
3.3.1 Översiktstabellen.....	9
3.3.2 Detaljtabellerna.....	11
3.3.3 Skapa diagram.....	11
3.4 Databasvård.....	12
3.4.1 Utföra en backup.....	12
3.4.2 Återställa en backup.....	12
3.4.3 Ta bort mätningar.....	13
4 Resultat.....	13
4.1 Synkronisering, överföring och verifiering.....	13
4.2 Applikationens grafiska gränssnitt och användarvänlighet.....	14
5 Diskussion.....	15
5.1 Databasmodellering.....	15
5.2 Överföra mätningar.....	16
5.3 Testning.....	16
6 Slutsatser.....	17
Referenser.....	18

1 Inledning

Inom Landstinget Gävleborg jobbar en forskningsgrupp med att utföra mätningar på förekomster av trycksår. Mätningarna utförs med hjälp av en surfplatta där resultatet sparas lokalt på surfplattans digitala minneskort. Examensarbetet består av att utveckla ett nytt datasystem som ska innehålla en databas med tillhörande databasapplikation. Systemet ska underlätta arbetet med att samla in mätningarna från surfplattan och lagra det i datasystemets databas. Applikationen ska kunna presentera resultatet från mätningarna på ett tydligt sätt.

1.1 Bakgrund

På Gävle sjukhus utförs mätningar på tryckskadeförekomster. I dagsläget görs det på 8 av sjukhusets avdelningar.

1.1.1 Teoretisk bakgrund

Punktprevalensmätning (PPM) av trycksår består av regelbundna stickprovskontroller som utförs genom att göra observationer, riskbedömningar och en journalgranskning på intagna patienter. Resultatet från dessa mätningar ska ligga till grund för att förebygga och minska förekomster av trycksår [1].

Trycksår är en lokal skada som kan uppstå på alla delar av kroppen. Mätningarna består av att göra en riskbedömning enligt den modifierade Nortonskalan. Om en person får 20 Nortonpoäng eller lägre löper denne risk att utveckla trycksår. På dessa personer ska det vidtas förebyggande åtgärder. Upptäcks ett trycksår vid en observation graderas det efter en fyrgradig skala [2].

PPM är en nationell satsning för att öka patientsäkerheten. Genom att få kontroll över frekvensen av trycksår får verksamheten ett underlag till ett långsiktigt, systematiskt och uthålligt förbättringsarbete. Vid ett mätningstillfälle 2011 hade Landstinget Gävleborg en total trycksårsfrekvens på 22 % och i hela Sverige var frekvensen 17 % [3].

1.1.2 Allmän bakgrund

På sjukhuset i Gävle utförs mätningar och stickprovskontroller på förekomster av trycksår och tryckskador på patienter. Detta ska göras en gång i veckan per avdelning och vid varje tillfälle tas ungefär 25 stickprov. En forskningsgrupp på Gävle sjukhus vill hitta effektiva och bra förebyggande åtgärder genom att utvärdera resultatet från mätningarna. Det ska ligga till grund för ett kvalitetsarbete som ska leda till att frekvensen och graden av trycksår minskar. Informationen från mätningarna samlas in med hjälp av surfplattor. Högskolan i Gävle har i samarbete med forskningsgruppen på Gävle sjukhus utvecklat en applikation som används på surfplattorna (se figur 1).

PPM-trycksår - Bedömningsprotokoll - Avdelning: qqq - 23 apr 2011 Spara och stäng

Bedömning Säng: 11.11 Födelseår: 1950

Persondata
 Kvinna Man
 Födelsedatum: 1 jan 1950
 Inskrivn på sjukhus: 22 apr 2011
 Inskrivn på avdelning: 23 apr 2011
 Prevention: Madrass: Förebyggande
 Annan utrustning: [0, 1, 1, 0, 0]
 Planerad lägesändring i säng: Nej
 Planerad lägesändring i stol: Nej

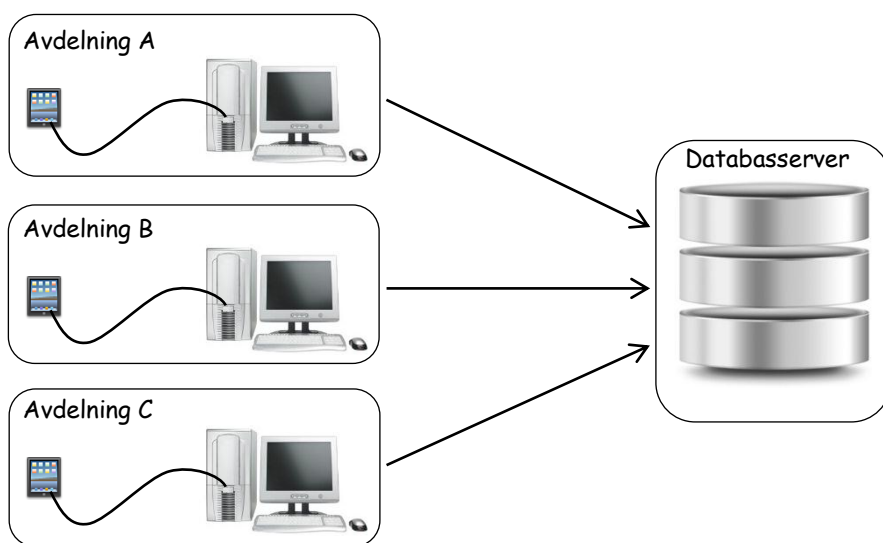
Journalgranskning
 Hudbedömning dokumenterad inom 24 timmar efter inskrivning på avdelningen Nej Ja
 Riskbedömning med riskbedömningsinstrument (Modifierad Nortonskala) dokumenterad inom 24 timmar efter inskrivning på avdelningen Nej Ja
 Trycksår/trycksador dokumenterade inom 24 timmar efter inskrivning på avdelningen Nej Ja
 Ingen skada Ingen skada

	Kat 1	Kat 2	Kat 3	Kat 4
Ryggslut	0 + 0	0 + 0	0 + 0	0 + 0
Häl	0 + 0	0 + 0	0 + 0	0 + 0
Fötter	0 + 0	0 + 0	0 + 0	0 + 0
Höft	0 + 0	0 + 0	0 + 0	0 + 0
Sittbensknölar	0 + 0	0 + 0	0 + 0	0 + 0
Öra	0 + 0	0 + 0	0 + 0	0 + 0
Öra pga. O2	0 + 0	0 + 0	0 + 0	0 + 0
Armbågar	0 + 0	0 + 0	0 + 0	0 + 0
Annat	0 + 0	0 + 0	0 + 0	0 + 0

Figur 1. PPM-trycksårapplikationen i surfplattorna där mätresultatet sparas i.

Applikationen lagrar resultatet från mätningarna i en inbäddad databas lokalt på surfplattan. Innehållet exporteras sedan till en tab-separerad textfil som sparas lokalt på surfplattans digitala minneskort. Eftersom det inte är tillåtet att använda trådlös kommunikation såsom WIFI eller Bluetooth på sjukhuset kan applikationen inte lagra data direkt till en central databasserver. För att överföra mätningarna från surfplattan måste personalen koppla den till en stationär dator på avdelningen via en USB-kabel (se figur 2).

I dagsläget öppnas den tab-separerade textfilen i Excel och innehållet sparas ner som en Excelfil. Sedan utförs arbetet av innehållet manuellt av personalen i Excel, men tanken är att i framtiden så ska alla avdelningars mätningresultat lagras i en central databasserver (se figur 2).



Figur 2. Surfplattan kopplas till en PC via en USB-kabel. I framtiden ska varje avdelnings mätningar lagras i en central databasserver.

1.2 Problembeskrivning

Eftersom det inte är tillåtet att använda trådlös kommunikation på surfplattorna så blir det problem att sammanställa och granska en avdelnings alla mätningar på ett och samma ställe. Den manuella hanteringen med att exportera innehållet till Excel fungerar i dagsläget inte så bra. Den tab-separerade textfilen är svårarbetad och saknar bra möjligheter att presentera innehållet utifrån olika sorterings- och filtreringsalternativ t.ex. efter ett visst sängnummer eller mätningstillfälle.

1.3 Avgränsning

Examensarbetet omfattar att skapa ett datasystem med tillhörande databas och applikation som skall köras på den stationära dator som finns på respektive avdelning (se figur 2). Systemet ska vara utvecklat för att köras i Windowsmiljö. Applikationen kommer att programmeras i Java Plattform, Standard Edition (SE) 6 [4] och distribueras som en Java Archive (JAR) fil som körs i Java Virtual Machine (JVM).

Databasen ska vara en inbäddad variant som heter ObjectDB. Det är en objektorienterad databashanterare anpassad för Java. ObjectDB kräver ingen licens om databasen har max 10 entitetklasser och inte överstiger en miljon entitetsobjekt [5]. ObjectDB kommer med Java Persistence API (JPA) version 2.0 [6], vilket är en inbyggd standard för relationsdatabaser. Skulle begränsningarna i ObjectDB överskridas går det bra att byta till en annan databashanterare i framtiden.

Design och överföring till den centrala databasservern där samtliga avdelningars mätningar ska lagras ingår inte i detta examensarbete.

1.4 Syfte

Syftet med detta examensarbete är att utveckla ett datasystem som kan samla in och importera mätningar från en surfplatta till en gemensam databas på en stationär dator. Systemets applikation ska innehålla olika presentationsverktyg som kan presentera databasens innehåll utifrån specifika kriterier. Målsättningen är att datasystemet ska underlätta arbetet med att samla in och presentera mätningresultatet, som ska ligga i grund till att förbättra kvaliteten och minska antalet tryckskadeförekomster och graden av trycksår.

1.5 Frågeställningar

Utvecklingen av datasystemet utgår från följande frågeställningar:

- Kan systemet synkronisera och importera mätningar från surfplattans databas via USB-porten och kontrollera så att inte blir dubbellagring av data i databasen?
- Kan systemet verifiera för användaren att överföringen gick korrekt till och ge en bekräftelse på hur många mätningar som har överförts?

- Kan applikationen utformas med ett användarvänligt grafiskt gränssnitt som kan presentera databasens innehåll utifrån användarens önskemål?
- Kan applikationen utvecklas så att den är enkel att använda och där utformningen anpassas utifrån att användarna endast har grundläggande datakunskaper?

2 Metod

Arbetsmetoden består av tre faser. Den inledande fasen är att titta på hur det ser ut i nuläget genom att utföra en verksamhetsanalys. När den är klar så kan bristerna identifieras och ligga till grund för systemets målsättningar. Den andra fasen består i att arbeta fram ett lösningsförslag och designa systemet utifrån de riktlinjer som satts upp i målsättningen. Den avslutande fasen är implementation av systemet efter den lösning och design man kommit fram till. Det ska leda till att målsättningarna uppfylls och lösa de problem verksamheten har idag.

2.1 Verksamhetsanalys

På sjukhuset i Gävle utför personalen riskbedömningar och mätningar på förekomster av trycksår och tryckskador på patienter. Detta skall utföras en gång i veckan per avdelning. I dagsläget utförs det kontroller på 8 avdelningar, men totalt kommer det att utföras stickprov på totalt 15-20 avdelningar. Vid varje mätningstillfälle ska det göras cirka 25 mätningar och till sin hjälp har personalen en surfplatta. På surfplattan finns det en applikation som innehåller ett bedömningsprotokoll som personalen använder för att dokumentera resultatet (se figur 1).

Applikationen sparar protokollen i en SQLite databas lokalt på surfplattan. Sedan exporteras mätningarna till en tab-separerad textfil som sparas på surfplattans digitala minneskort. Eftersom trådlös kommunikation inte är tillåtet så måste surfplattan kopplas till en stationär dator via USB-porten för att överföra innehållet till sjukhusets nätverk.

Mätningarna och bedömningsprotokollen behandlas i dagsläget manuellt av en person. Den tab-separerade textfilen öppnas i Excel och varje avdelning får sin egen Excelfil som resultatet sparas i. Där klipps och klistras innehållet ihop manuellt till lämpliga tabeller. Mätvärdena beräknas, analyseras och presenteras på kalkylblad och med olika diagram som finns i Excel.

Syftet med dessa mätningar och stickkontroller är att utreda hur många trycksår som har uppkommit sedan en patient skrevs in på respektive avdelning. Det görs genom att jämföra stickprovet med vad som tidigare fanns dokumenterat i patientens journal vid inskrivningstillfället på andelningen. Fanns det ingen skada eller att det saknas dokumentation om trycksår i journalen, så ska eventuella uppkomna trycksår tolkas som att det är orsakade på den avdelning patienten är inlagd på.

Målsättningen med resultatet från mätningarna är att de ska ge stöd åt personalen och ligga till grund för ett kvalitetsarbete med att förbygga uppkosten av trycksår. Genom att analysera och utvärdera de förebyggande åtgärder och utrustning som

används, så kan man hitta det som är mest effektivt och på så sätt styra kommande arbetsinsatser för att minska frekvensen och uppkomsten av trycksår.

2.2 Problemanalys

Eftersom trådlös kommunikation inte är tillåtet så skapas det ett problem med att bedömningsprotokollen inte kan lagras direkt till en central databasserver på sjukhusets nätverk, utan det sparas lokalt på surfplattans digitala minneskort.

Den tab-separerade textfilen som mätningarna exporteras till är svårarbetad. När den öppnas i Excel får man upp innehållet på drygt 100 kolumner. Kolumnnamnen är svåra att tyda och det är svårt att veta vilken information respektive kolumn egentligen innehåller.

Det manuella arbetet i Excel med att klippa och klistra är inte effektivt och risken för att det blir fel ökar. När antalet mätningar blir fler, så leder det till att arbetet både blir svårare och mer tidskrävande.

2.3 Funktionsanalys

En funktionsanalys gjordes för att konkret bestämma vilka funktioner systemet ska kunna utföra. Utifrån det kan man sedan skapa tydliga målsättningar baserat på de absolut viktigaste funktionerna systemet ska uppfylla. Det är till stor hjälp och stöd för utvecklingen av systemet och det ger tydliga riktlinjer för hur arbetsprocessen ska läggas upp.

Dessa målsättningar sattes upp för systemet:

1. Systemet skall kunna synkronisera och överföra mätningar från surfplattan till en stationär dator utan att det blir dubbellagring av data i databasen.
2. Applikationen ska kunna presentera mätningarna både översiktligt och detaljerat.
3. Det ska gå att skapa enklare linjediagram på hur många trycksår som har uppkommit sedan inskrivning på avdelningen jämfört med det som finns dokumenterat i journalen. Diagrammet ska visa hur antalet trycksår varierar i procent över en viss tidsperiod.
4. Applikationen ska ha ett verktyg för att exportera hela databasen till ett lämpligt filformat.
5. Applikationen ska ha funktioner för allmän databasvård, såsom säkerhetskopiering, återställning och en funktion för att ta bort mätningar i databasen.

2.4 Lösningsförslag

I framtiden är det tänkt att mätningarna ska utföras av respektive avdelning. För att samla in och lagra en avdelnings mätningar ska en ny databas och applikation skapas. Denna applikation ska köras på en stationär dator och varje avdelning får sin egen databas att samla in sina mätningar på. Efter varje mätningstillfälle ska surfplattan kopplas till datorn med en USB-kabel (se figur 2). Programmet ska ha en funktion som kan synkronisera och överföra mätningarna mellan surfplattan och datorn. Det ska också ingå en kontroll i överföringen att ingen dubbellagring av data sker i den nya databasen.

Applikationen ska också ha ett verktyg för att visa databasens innehåll i form av en översiktstabell. Denna tabell ska skapas utifrån avdelning och vilket datum mätningstillfället utfördes på. Tabellen ska visa alla sängar från det valda mätningstillfället. Till varje säng ska patientens kön, födelseår, summan från Nortonskalan, riskbedömning och antalet uppkomna trycksår visas. Det ska vara lätt för personalen att se vilka sängar som har orsakat trycksår och som är uppkomna på avdelningen.

Vill personalen titta närmare på en säng och få en mätningens fullständiga detaljer så ska applikationen också ha ett verktyg för det. Varje mätning får sitt unika ID i databasen och det framgår i översiktstabellen vilket ID varje mätning har. Genom att ange mätningens ID så ska applikationen hämta all information som finns lagrat på det ID.

Programmet ska också kunna skapa diagram. Det ska vara ett linjediagram som visar hur många trycksår i procent som uppkommit på mätningstillfällena under en viss tidsperiod. Detta ska vara avdelningsvis och diagrammet skall ha två kurvor. Den ena kurvan visar uppkomna trycksår på avdelningen och den andra kurvan visar hur det såg ut vid inskrivningstillfället. Syftet är att det ska tydligt framgå skillnaden mellan hur många trycksår som uppkommit på avdelningen från patientens inskrivningsdatum. Man ska också kunna analysera topparna i diagrammet så man kan hitta de faktorer som orsakade en ökning av trycksår vid något mätningstillfälle. En orsak skulle t.ex. kunna vara att beläggningen av patienter var högre än normalt och det ledde till att arbetet med förebyggande åtgärder blev lidande.

Export av databasen kommer att ske genom att överföra databasens innehåll till en tab-separerad textfil. Exportfunktionen kommer också fungera som backup och säkerhetskopiering av databasen. En funktion för import och återställning av systemets databas kommer att ingå tillsammans med radering av mätningar i den allmänna databasvärden.

3 Genomförande

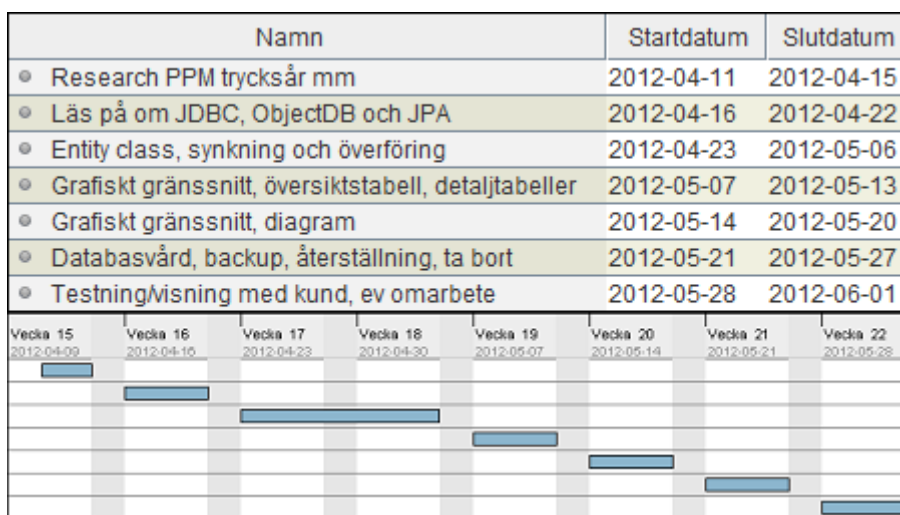
För att få en bra överblick och förståelse över hur verksamheten fungerar i dagsläget utfördes en nutidsanalys. Verksamheten kartlades för att identifiera dess problem. Genom att ta reda på vilka funktioner systemet skall utföra kan man skapa tydliga målsättningar som datasystemet ska uppfylla. Det ger ett bra stöd och riktlinjer för att planera genomförandet av arbetsgången, som man sedan kan följa i utvecklingen, designen och implementationen av datasystemet.

3.1 Planering

Efter att fått projektbeskrivningen godkänd påbörjades projektarbetet. För att kunna göra en bra planering gjordes en research utifrån projektbeskrivningen. Det handlade om att läsa in sig på det ämne projektet bestod av, tryckskadeförekomster inom sjukvården. Det är viktigt att bilda sig en uppfattning om det ämne projektet handlar om för att kunna göra ett bra jobb.

Nästa steg var att sätta sig in i det programmeringsspråk datasystemet skulle programmeras i. Kunskaper inom Java och relationsdatabaser fanns sedan tidigare, men det som var nytt var att arbeta med dessa två tillsammans. Framförallt var det att läsa på om grunderna i Java Database Connectivity (JDBC) [7], som är till för att Java ska kunna kommunicera med databasen i surfplattan. Samma sak gjordes med JPA [6] som ObjectDB använder. En repetition i Java med om hur man jobbar med grafiska komponenter samt läsning och skrivning till textfiler gjordes också [8].

Då den inledande delen var avslutad gjordes en planering utifrån projektbeskrivning och frågeställning som satts upp (se figur 3).



Figur 3. Planering gjord i verktyget GanttProject.

Efter att den tekniska biten med att lösa synkronisering och överföring från surfplattans databas till systemets databas var klar bokades ett möte in med kunden. På det mötet sattes det upp riktlinjer för hur det grafiska gränssnittet skulle utformas utifrån kundens önskemål. Med på mötet var också en statistiker som redogjorde för hur beräkningarna för diagrammet skulle utföras.

3.2 Synkronisera och överföra mätningar

En av systemets huvudfunktioner är att synkronisera, överföra och verifiera överföringarna mellan surfplattans och applikationens databas med en USB-kabel (se figur 2).

3.2.1 Surfplattans databas

Det första som gjordes var att analysera SQLitedatabasen i surfplattan för att ta reda på vilka attribut och datatyper objektet hade. SQLite kräver ingen server som de flesta andra SQLdatabaser gör, utan läser och skriver direkt till vanliga filer [9]. För att en Javaapplikation ska kunna kommunicera med SQLitedatabasen behövs JDBC och i detta system användes en driver som finns i ett JAR av typen SQLiteJDBC v056 [10]. För att hantera kommunikationen med surfplattans databas skapades en ny klass som döptes till TabletDB. Den innehåller metoder för uppkoppling, nedkoppling och en instruktion som returnerar en SQL-fråga från inmatad SQL-sats (se figur 4).

```
public ResultSet makeQuery(String s) throws SQLException {  
    return statement.executeQuery(s);  
}
```

Figur 4. En metod som returnerar en SQL-fråga i klassen TabletDB.

3.2.2 Datasystemets databas

Datasystemets databashanterare heter ObjectDB, och är en objektorienterad och inbäddad variant anpassad för Java [5]. ObjectDB använder sig av JPA [6] och det finns i ett JAR som läggs till i projektet.

Det skapades en ny klass med namnet ObjectDB med metoder för att skapa upp- och nedkoppling mot databasen samt för att lagra och ta bort poster i databasen. Dessutom innehåller klassen metoder med instruktioner för att returnera en SQL-fråga eller en lista med strängar utifrån en SQL-sats.

3.2.3 Lagra data i ObjectDB

För att lagra data i applikationens databas skapades en entitetklass. Namnet på den klassen är Measurement och för att deklarerar den som en entitetklass använder man en annotering som är @Entity. Entitetklassen innehåller variabler för varje attribut som är av lämpliga datatyper baserade på dess innehåll. En variabel motsvarar en kolumn i databasen och varje variabel har sina egna get- och setmetoder i klassen. För varje mätning som ska lagras i databasen skapas ett nytt objekt av entitetklassen och varje attribut tilldelas den information som ska lagras med hjälp av respektive setmetod.

3.2.4 Importera mätningar från surfplattan

Överföring och import av mätningar görs av klassen ImportFromTablet. Den klassen skapar uppkopplingar både mot surfplattans och mot datasystemets databaser. Klassen hanterar också exceptions om systemet misslyckas med att få kontakt med någon av databaserna och meddelar användaren på skärmen. Innan en ny mätning kan importeras görs en dubbellagringskontroll. Varje mätning har ett universallt unikt identifier (UUID), så innan en ny mätning överförs, jämförs det nya UUID med de redan lagrade UUIDs i databasen (se figur 5).

```

// select all from TabletDB
resultSet = tabletDB.makeQuery("SELECT * FROM assessments");
// if DB has content, put all uuid's in a List<String>
if (emptyDB == false) {
    uuid = objectDB.stringStatementObjectDB(
        "SELECT DISTINCT NEW String(m.uuid) FROM Measurement m");
}
// loop through tabletDB query
while(resultSet.next()) {
    uuid_already_stored = false; // reset
    // if DB not empty...
    if (emptyDB == false) {
        // check if uuid is stored in DB already
        new_uuid = resultSet.getString("uuid");
        for (int i=0 ; i<uuid.size() ; i++) {
            stored_uuid = uuid.get(i);
            if(new_uuid.equals(stored_uuid)) {
                // set true if uuid already in DB
                uuid_already_stored = true;
            }
        }
    }
    // if uuid not in DB, store it
    if(uuid_already_stored == false) {
        addMeasurement();
        count++; // counter
    }
}
}

```

Figur 5. Dubbellagringskontroll innan en ny mätning kan importeras.

Om en mätning inte finns lagrad i databasen anropas en metod för att hämta data från surfplattans databas och sätta in den i respektive attribut i entitetsklassen. Då alla attribut är tillagda skapas ett nytt entitetsobjekt som lagras i systemets databas. Varje gång en ny mätning läggs till i databasen räknar en räknare upp. Metoden avslutas med att den returnerar räknaren och meddelar användaren om hur många nya mätningar som har lagts till i databasen.

3.3 Presentation av mätningar

Presentationen av resultatet från mätningarna ska kunna göras både översiktligt och detaljerat. Huvudklassen för det grafiska användargränssnittet är StartApp. Den klassen innehåller alla de grafiska komponenter som behövs för att jobba med databasens innehåll.

3.3.1 Översiktstabellen

Syftet med översiktstabellen är att användaren ska kunna lista mätningarna avdelningsvis och vilket datum de utfördes på. Översiktstabellens innehåll har designats utifrån personalens önskemål och genom att titta i tabellen ska man snabbt kunna se vilka mätningar som har trycksår (se figur 6).

SKAPA ÖVERSIKTSTABELL Avdelning: Avd.A Datum: 2012-05-18

Listan innehåller 12 st mätningar

ID	Säng	Kön	Födelseår	Norton	Risk	Trycksår
265	11.11	Man	1945	22	Ingen risk	0
266	11.12	Man	1955	17	Risk	1
267	11.13	Kvinna	1966	18	Risk	0
268	11.14	Kvinna	1955	16	Risk	3
269	11.15	Man	1925	16	Ingen risk	0
270	11.16	Kvinna	1987	21	Ingen risk	2
271	11.17	Kvinna	1963	20	Risk	0
272	11.18	Man	1990	17	Risk	0
273	11.19	Man	1956	21	Ingen risk	0
274	11.20	Man	1935	14	Risk	0
275	11.21	Kvinna	1942	18	Risk	0
276	11.22	Kvinna	1974	18	Risk	0

Figur 6. Tabellen visar mätningar från den 18 maj, 2012 på Avd.A.

Det finns också ett alternativ att lista alla mätningar på en avdelning som sorteras efter sängnummer. Syftet med det är att personalen enkelt kan följa utvecklingen på en specifik säng i flera mätningar.

Det är totalt tre klasser som används för att skapa översiktstabellen. Combo-boxarna anropar metoder i klasserna WardList och DateList. Dessa två metoder returnerar en array med strängar som innehåller avdelningarnas namn eller mätningstillfällenas datum (se figur 7).

```
public String[] getWardsList(String url) {
    objectDB.connectObjectDB(url);
    List<String> wards = null;
    try {
        wards = objectDB.stringStatementObjectDB("SELECT DISTINCT
            NEW String(m.ward) FROM Measurement m ORDER BY m.ward");
        wards.add(0, "Välj avdelning");
    }
    catch (PersistenceException pe) {
        System.err.println("Error reading Measurement wards");
        wards = new ArrayList<String>();
        wards.add(0, "Inga avdelningar");
    }
    finally {
        objectDB.closeObjectDB();
    }
    return wards.toArray(new String[wards.size()]);
}
```

Figur 7. En metod som returnerar en array med strängar på avdelningarnas namn.

Själva tabellen skapas av klassen OverviewTable. Den skapar en uppkoppling till databasen och genom att anropa metoden makeOverviewTable hämtas data för att fylla upp tabellens kolumner och rader.

3.3.2 Detaljtablerna

För att titta närmare på ett mätningresultat kan man ange mätningens ID-nummer för att visa dess detaljer. ID-numret finns med som en kolumn i översiktstabellen (se figur 5). Detaljtablerna skapas av klassen DetailTable. Den skapar en uppkoppling till databasen och genom att anropa metoden makeDetailTables skapas tabellerna om det ID man angav finns i databasen (se figur 8). Det går bara att ange siffror i textfältet, så skulle användaren försöka mata in något annat eller ett ID som inte finns i databasen får denne ett varningsmeddelande.

SKAPA DETALJTABELLER		Ange ID: 266	Visa detaljer			
Bedömningsprotokoll		Trycksår avdelning				
ID	266		Kat 1	Kat 2	Kat 3	Kat 4
Datum	2012-05-18	Ryggslut	0	1	0	0
Avdelning	Avd.A	Häl	0	0	0	0
Säng	11.12	Fötter	0	0	0	0
Kön	Man	Höft	0	0	0	0
Födelseår	1955	Sittbensknölar	0	0	0	0
Inskrivn på sjukhus	15-mar-2012	Öra	0	0	0	0
Inskrivn på avdelning	16-apr-2012	Öra pga O2	0	0	0	0
		Armbågar	0	0	0	0
		Annat	0	0	0	0

Figur 8. Två stycken av ID-nummer 266 detaljtabeller.

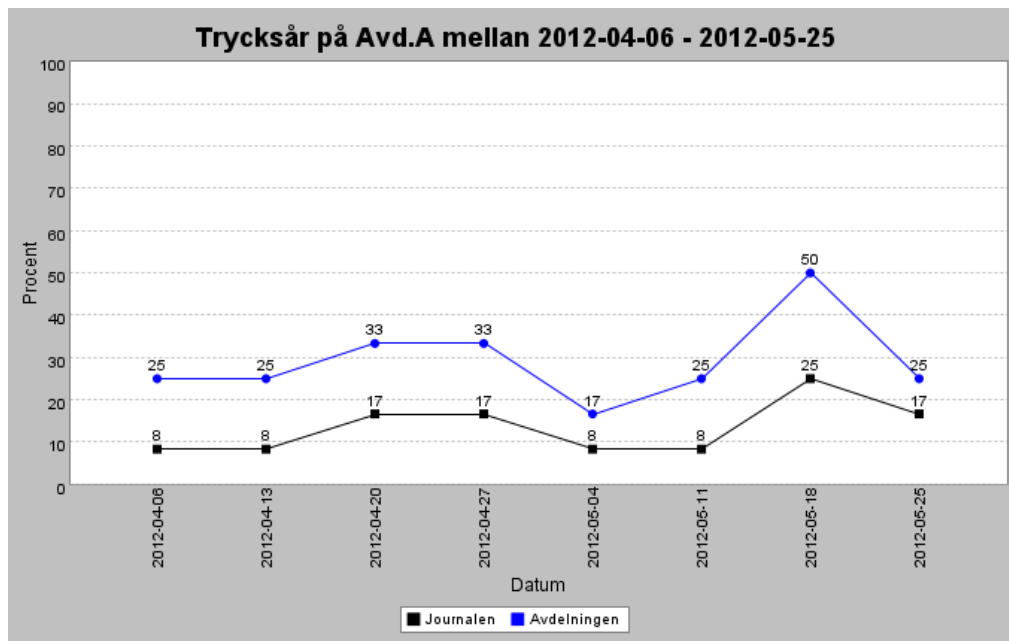
3.3.3 Skapa diagram

Ett verktyg som applikationen ska ha är möjligheten att skapa diagram. Längst ner i applikationen finns ett fält för att ställa in vilken avdelning och mellan vilka mätningstillfällen som diagrammet ska visa (se figur 9).

SKAPA DIAGRAM	Avdelning: Avd.A	från: 2012-04-06	till: 2012-05-25	Skapa diagram
---------------	------------------	------------------	------------------	---------------

Figur 9. Verktygsfältet för att skapa diagram.

Det linjediagram som skapas ska visa två stycken kurvor. Den ena kurvan visar hur många trycksår som var dokumenterade i journalen då patienten skrevs in på avdelningen och den andra kurvan ska visa hur många trycksår som hittades vid mätningstillfället. Resultatet ska räknas om och redovisas i procent över den tidsperiod som har angetts (se figur 10).



Figur 10. Diagram på Avd. A från 2012-04-06 till 2012-05-25.

För att skapa diagrammet användes ett open source Javabibliotek som heter JFreeChart [11]. Biblioteket kommer med ett väldokumenterat API [12] innehållande många olika diagrammodeller och de två JARs applikationen använder är jcommon-0.6.3 och jfreechart-0.9.1.

3.4 Databasvård

Applikationen innehåller en allmän databasvård med funktioner såsom backup, återställning och möjligheten att radera data ur databasen.

3.4.1 Utföra en backup

Systemet har ett enkelt och snabbt verktyg för att göra en backup och exportera databasens innehåll. Mätningarna sparas ner i en tab-separerad textfil med klassen BackupExport. De tab-separerade filerna sparas i en mapp med datum och timestamp i filnamnet. Den tab-separerade filen kan t.ex. öppnas i Excel eller användas för att exportera mätningar till den centrala databasservern i framtiden (se figur 2). Klassen hanterar också exceptions för behörighetskontroll till filen eller att ett fel uppstod då programmet försökte öppna eller skriva till filen och meddelar användaren på skärmen.

3.4.2 Återställa en backup

För att kunna återställa systemet från en exporterad backupfil så skapades klassen RestoreBackupExport. Den klassen innehåller en scanner för att kunna scanna igenom backupfilen radvis och lägga det i en sträng. Sedan skapas ytterligare en scanner för att scanna ord för ord på varje linje. Det utförs med nästlade while loopar (se figur 11) och innan en import till databasen görs utförs också en dubbellagringskontroll.


```

// scan the file every line
Scanner scanner = new Scanner(file);
scanner.nextLine(); // don't add column names
while (scanner.hasNextLine()) {
    // scan every line and store it in a string
    String line = scanner.nextLine();
    Scanner lineScanner = new Scanner(line);
    while (lineScanner.hasNext()) {
        ...
    }
}

```

Figur 11. Nästlade while loopar för att scanna backup filen.

3.4.3 Ta bort mätningar

Eftersom ObjectDB har vissa begränsningar så måste möjligheten att ta bort mätningar från databasen finnas. Detta görs med klassen DeleteEntity. Mätningar tas bort genom att användaren anger ett datum och alla mätningar äldre fram till och med det datumet raderas ur databasen.

4 Resultat

Applikationen som har utvecklats har döpts till PPM ShowTool och innehåller olika funktioner att överföra och presentera mätningar samt en allmän databasvård.

4.1 Synkronisering, överföring och verifiering

Datasystemet kan synkronisera och överföra mätningarna från en SQLitedatabas till systemets ObjectDB databas. Däremot går det i dagsläget inte att komma åt surfplattans databas direkt med USB-kabeln på grund av att den lagras på surfplattans interna minne, så när surfplattan kopplas in via USB-porten kommer man endast åt surfplattans digitala minneskort. För att denna funktion ska fungera måste SQLite-databasen lagras direkt på surfplattans digitala minneskort alternativt kunna exporteras dit manuellt via applikationen i surfplattan.

Innan varje mätning överförs till databasen utförs en dubbellagringskontroll (se figur 5). Innan en mätning läggs till i databasen jämförs det UUID varje mätning har fått då den skapades i surfplattan. Efter att överföringen är klar får användaren ett meddelande om att överföringen lyckades och hur många mätningar som har överförts till databasen (se figur 12). Det blir alltså ingen dubbellagring av data i databasen.



Figur 12. Bekräftelse på att databasen har uppdateras.

4.2 Applikationens grafiska gränssnitt och användarvänlighet

Det grafiska gränssnittet har skapats för att vara lätt att använda och innehållet har tagits fram i samtal med kund. Det har varit viktigt att ta del av kundens önskemål i utvecklingen av programmet. Kommunikationen med kunden har skett både via kundbesök och via epost.

Varje presentationsverktyg har sin egen del på verktygsfälten och på visningsytan. Det framgår tydligt vilken del som gör vad med hjälp av beskrivande rubriker. För att det ska gå enkelt och snabbt att ta fram en översiktstabell eller ett diagram används comboboxar som uppdateras automatiskt så fort användaren ändrar någon parameter.

Det grafiska gränssnittet är uppdelat med två stycken verktygsfält och en visningsyta för översikt- och detaljtabellerna (se figur 13). I det övre fältets vänstra sida skapar man en översiktstabell genom att välja avdelning och mätningsdatum (se figur 6). För att göra det enkelt för användaren används en combo box där alla avdelningar som finns lagrade i databasen listas. Då användaren har valt en avdelning uppdateras den andra combo boxen med de mätningsdatum som finns lagrade på den avdelningen och sedan skapas tabellen så fort datumet har valts.

ID	Säng	Kön	Födelseår	Norton	Risk	Trycksår
244	11.14	Man	1932	17	Risk	1
256	11.14	Man	1992	21	Risk	0
268	11.14	Kvinna	1955	16	Risk	3
185	11.15	Man	1943	17	Risk	0
197	11.15	Man	1945	20	Risk	1
209	11.15	Man	1959	19	Ingen risk	0
221	11.15	Man	1929	19	Risk	0
233	11.15	Kvinna	1946	15	Risk	1
245	11.15	Man	1918	20	Risk	0
257	11.15	Man	1923	19	Risk	0
269	11.15	Man	1925	16	Ingen risk	0
186	11.16	Man	1933	23	Ingen risk	0
198	11.16	Man	1989	22	Risk	0
210	11.16	Kvinna	1944	23	Ingen risk	0
222	11.16	Kvinna	1971	20	Risk	0
234	11.16	Man	1938	24	Risk	0
246	11.16	Man	1976	19	Risk	0
258	11.16	Man	1963	22	Risk	0
270	11.16	Kvinna	1987	21	Ingen risk	2
187	11.17	Kvinna	1976	23	Ingen risk	0
199	11.17	Kvinna	1944	21	Ingen risk	0
211	11.17	Kvinna	1954	23	Risk	0
223	11.17	Kvinna	1978	20	Risk	0
235	11.17	Kvinna	1969	24	Ingen risk	0
247	11.17	Kvinna	1987	20	Ingen risk	0
259	11.17	Kvinna	1986	22	Ingen risk	0
271	11.17	Kvinna	1963	20	Risk	0

ID	Datum	Avdelning	Säng	Kön	Födelseår	Inskrivna på sjukhus	Inskrivna på avdelning
270	2012-05-18	Avd.A	11.16	Kvinna	1987	14-apr-2012	4-maj-2012

Trycksår avdelning	Kat 1	Kat 2	Kat 3	Kat 4
Ryggslut	0	0	0	0
Häl	0	0	0	0
Fötter	0	0	0	0
Höft	0	0	0	0
Sittbensknölar	1	0	0	0
Öra	0	0	0	0
Öra pga O2	0	0	0	0
Armbågar	1	0	0	0
Annat	0	0	0	0

Journalgranskning	Hudbedömning dokumenterad	Riskbedömning Nortonskala	Trycksador dokumenterade
	Nej	Ja	Nej

Trycksår avdelning	Kat 1	Kat 2	Kat 3	Kat 4
Ryggslut	0	0	0	0
Häl	0	0	0	0
Fötter	0	0	0	0
Höft	0	0	0	0
Sittbensknölar	0	0	0	0
Öra	0	0	0	0
Öra pga O2	0	0	0	0
Armbågar	0	0	0	0
Annat	0	0	0	0

Figur 13. En bild på hur hela applikationen PPM ShowTool ser ut.

Översiktstabellens kolumner är framtagna så att det enkelt och snabbt syns vilka mätningar som riskerar att utveckla trycksår eller där trycksår redan har upptäckts. Det går också att skapa en översiktstabell med samtliga mätningar från en avdelning som sorteras efter sängnummer. Det är bra då personalen vill kunna följa en och samma säng under en längre period (se figur 13).

På den högra sidan av verktygsfältet kan användaren visa alla detaljer på en mätning (se figur 8). Det görs genom att ange det unika ID nummer varje mätning har fått i databasen. Skulle användaren ange ett ID som inte finns lagrat i databasen eller skriva något annat än siffror i textfältet visas ett felmeddelande. Här finns det ett

utrymme för en förbättring. En idé skulle kunna vara att lägga till en funktion som tillåter användaren att navigera i översiktstabellen och så känner systemet av vilken mätning som är markerad och uppdaterar informationen i detaljtabellerna automatiskt.

Utformningen av detaljtabellerna (se figur 13) gjordes utifrån hur applikationen i surfplattan är designad (se figur 1). Detaljtabellernas uppdelning och innehåll ska kunna kännas igen från bedömningsprotokollet. Syftet med det är att det ska vara lätt för användaren att känna igen sig i miljön från surfplattan och enkelt kunna förstå informationen i detaljtabellerna.

I det nedre verktygsfältet kan användaren skapa ett diagram (se figur 9). Det går att skapa diagram avdelningsvis eller ett diagram över alla avdelningar som finns lagrade i databasen. Sedan anger man mellan vilka datum diagrammet ska visa. Comboboxarnas datum uppdateras utifrån vilken avdelning som är vald. Det är också programmerat så att slutdatumet i diagrammet inte kan vara lika med eller före startdatumet. Efter att ha angett avdelning, från- och tilldatum i comboboxarna skapas diagrammet genom att klicka på knappen ”skapa diagram”.

Diagrammet öppnas i ett eget fönster (se figur 10) och det kan maximeras på skärmen om så önskas. Diagrammet har två stycken kurvor, den svarta kurvan visar om det fanns trycksår dokumenterade i patientens journal vid inskrivning på avdelningen och den blå kurvan visar de trycksår som upptäcktes vid mätningstillfället. Syftet med diagrammet är att det ska tydligt framgå hur frekvensen i procent av trycksår har förändrats sedan patienten skrevs in på avdelningen.

5 Diskussion

I den här delen ska jag ge en del personliga reflektioner och diskutera kring några delar av projektet som inte gick som det var planerat ifrån början.

5.1 Databasmodellering

Den databas som finns i surfplattan sparar all data i en och samma databastabell. Så i början av projektet var utgångspunkten att det nya datasystemets databas skulle modelleras. Jag gjorde även ett modelleringsförslag, men efter ett tag insåg jag svårigheten med att hitta bra unika nycklar. Mätningen självt är ju ett objekt och den ska innehålla mätningresultatet. Eftersom patienten i sig inte är av intresse i mätningen så lagras inte något personnummer eller patientnummer i mätningen och då kan inte patienten bli ett objekt.

Det finns dock två möjliga objekt och det är avdelningen och sängen. Problemet med dessa är att de namnges manuellt vid varje mätningstillfälle och det är den enda information som finns tillgänglig och att skapa unika nycklar utifrån det känns inte optimalt.

En lösning på problemet skulle vara att applikationen i surfplattan har en möjlighet att lägga in de avdelningar mätningar utförs på. När nya mätningar ska utföras väljer man den avdelning mätningen görs på, utav de som är inlagda sedan tidigare. Nästa steg efter det är att ange vilka sängar som tillhör vilken avdelning och då får man en unik sammansatt nyckel.

Fördelen med att ha bara ett objekt i en ObjectDBdatabas är att det blir bara en entitetsklass och då blir det bara ett entitetsobjekt per mätning, vilket betyder att det går att lagra en miljon mätningar innan gränsen för att använda databasen gratis överskrids.

5.2 Överföra mätningar

Kommunikationen mellan surfplattan och en vanlig dator med en USB-kabel är inte helt problemfri. Själv hade jag inte tillgång till en surfplatta och PPM-trycksårapplikationen förrän några dagar i slutet av projektet. Under utvecklingen av datasystemet har jag använt en USB-sticka istället där en kopia av surfplattans SQLitedatabas funnits på. För att synkronisering och överföring ska kunna ske direkt från surfplattan måste det göras ändringar i applikationen på surfplattan så att SQLite-databasen blir tillgänglig.

För att ändå kunna överföra mätningar från surfplattan skapades i sista stund en importfunktion som kan importera mätningarna från den exporterade tab-separerade textfilen istället. Den ligger tillgänglig på surfplattans digitala minneskort och kan användas för att överföra mätningar till datasystemets databas. Tyvärr var inte heller den metoden helt problemfri.

När personalen utför mätningarna på sjukhuset sparas vissa mätningar som ofullständiga. Detta är på grund av att mätningarna är frivilliga och alla patienter vill inte delta. Dessa patienters mätningar sparas som ofullständiga. När en ofullständig mätning exporteras skapas det blanka attributfält i den tab-separerade textfilen och överföringen avbryts. Det man kan göra är att utföra och spara de fullständiga mätningarna först så de ofullständiga hamnar sist i textfilen. Då importeras de fullständiga mätningarna till databasen och avbryts då den första ofullständiga mätningen kommer. Detta har verifierats att det fungerar på riktiga mätningar.

Personalen är medveten om problemet och ska tänka på att lägga de ofullständiga mätningarna sist i mätningssamlingen. En annan diskussion är om ofullständiga mätningar ska in i systemet överhuvudtaget och hur är det tänkt att hantera dessa med tanke på uträkningar och statistik? På den frågan finns inget svar i dagsläget.

Det som kan anses vara mest avancerat är att användaren själv måste öppna den fil där databasen eller textfilen finns. Användaren måste alltså vara bekant med hur man navigerar i datorns enheter med flyttbara lagringsmedia för att hitta rätt sökväg. En optimal lösning i framtiden skulle kunna vara att systemet själv ska kunna känna av att surfplattan kopplas in till datorn och lokalisera filen. Sedan behöver användaren bara klicka på en knapp för att bekräfta så överförs mätningarna.

5.3 Testning

På grund av strul med drivrutiner har surfplattan inte kunnat kopplas till sjukhusets datorer så att kommunikationen fungerar. Det betyder att datasystemet inte kunnat testas ordentligt. På grund av detta är det svårt att analysera om utformningen av applikationen är anpassad utifrån att användarna endast har grundläggande datakunskaper.

Under utvecklingen har jag testat och verifierat systemet allt eftersom med påhittade mätningar lagrade i en kopia av surfplattans databas på en USB-sticka. Detta är dock inte tillräckligt och systemet bör testas på riktigt innan det tas i bruk.

6 Slutsatser

I den här delen redovisas frågeställningarna som sattes upp i inledningen:

- Kan systemet synkronisera och importera mätningar från surfplattans databas via USB-porten och kontrollera så att inte blir dubbellagring av data i databasen?
I dagsläget går det inte att importera mätningar direkt från surfplattans databas utan att vissa ändringar görs i surfplattans applikation. Data-systemets dubbellagringskontroll fungerar och ingen mätning registreras mer än en gång i databasen.
- Kan systemet verifiera för användaren att överföringen gick korrekt till och ge en bekräftelse på hur många mätningar som har överförts?
Användaren får ett meddelande att överföringen var lyckad och hur många mätningar som har överförts.
- Kan applikationen utformas med ett användarvänligt grafiskt gränssnitt som kan presentera databasens innehåll utifrån användarens önskemål?
Innehållet i det grafiska gränssnittet har tagits fram i samtal med kund och den feedback kunden har gett under utveckling, visning och testning av programmet har varit positiv.
- Kan applikationen utvecklas så att den är enkel att använda och där utformningen anpassas utifrån att användarna endast har grundläggande datakunskaper?
Detta var svårt att analysera eftersom det inte var möjligt att testa programmet med flera olika användare på sjukhuset på grund av strul med drivrutiner för att få surfplattan att fungera korrekt på sjukhusets datorer.

Referenser

- [1] Sveriges kommuner och landsting, *instruktioner för trycksårsmätning*, 2012.
http://www.skl.se/BinaryLoader.axd?OwnerID=0d739c0f-f65d-4487-99a5-232b4fc8a60f&OwnerType=0&PropertyName=EmbeddedFile_1506af5e-cbe4-4605-ac42-08e7bea9c129&FileName=Instruktioner+PPM-trycks%c3%a5r+20120109.pdf&Attachment=True (besökt 2012-04-12)
- [2] Vårdhandboken, Inera AB, Stockholm, 2011.
<http://www.vardhandboken.se/Texter/Trycksar/Oversikt/> (besökt 2012-04-12)
- [3] B. Thelberg, *Nationell trycksårsmätning i Landstinget Gävleborg v 12*, 2011
http://www.lg.se/Global/Jobba_med_oss/vargivarportalen/Patientsakerhet/nationell_trycksarsmatning_v_12_2011_rapport.pdf (besökt 2012-05-21)
- [4] Java™ Platform, Standard Edition 6 API Specification, Oracle, 2006.
<http://docs.oracle.com/javase/6/docs/api/> (besökt 2012-04-16)
- [5] ObjectDB Object Oriented Database Management System (ODBMS). ObjectDB Software, 2010. <http://www.objectdb.com> (besökt 2012-04-16)
- [6] JPA 2.0, <http://docs.oracle.com/javaee/5/api/javax/persistence/package-summary.html><http://sqlite.org/about.html> (besökt 2012-04-16)
- [7] Paul Deitel and Harvey Deitel, *Java How to Program*. Prentice Hall, 2012.
- [8] J. Skansholm, *Java direkt med Swing*. Studentlitteratur, Lund, 2005.
- [9] SQLite, <http://www.sqlite.org/about.html> (besökt 2012-04-16)
- [10] David Crawshaw, <http://www.zentus.com/sqlitejdbc/> (besökt 2012-04-16)
- [11] JFreeChart, <http://www.jfree.org/index.html> (besökt 2012-05-17)
- [12] JFreeChart API, <http://www.jfree.org/jfreechart/api/javadoc/index.html> (besökt 2012-05-17)