

UNIVERSITETET I BERGEN

INSTITUTT FOR INFORMASJONS- OG MEDIEVITENSKAP

**Information system for postmarket
surveillance of total joint prostheses**

Master thesis



Author:

Eirik BERNTSEN

Supervisor:

Ankica BABIC

May 31, 2014

Abstract

The storage, integration and presentation of clinical data is an important aspect of any modern medical research. The Biomaterials research group at the Haukeland University Hospital uses both their own locally generated clinical data and external registry data to examine retrieved joint implants. This thesis describes the design and implementation of a database and an application for use at the Biomaterials lab. Its goal was to enable an integrated storage and retrieval system that would combine information from different data sources used by the researchers.

As a solution to this challenge, a system prototype was developed that would enable further integration of these information systems into a multi-user environment. The system allows importing registry data and matching it with local data, viewing and editing of this information and exporting the integrated data for further statistical analysis.

To guide the research project and improve traceability, the design science paradigm was followed. For the development phase, elements from agile methodologies such as user stories and sprints were applied.

An evaluation consisting of both user testing and heuristic evaluation was carried out and generated constructive feedback and positive responses.

The prototype demonstrates the feasibility of combining these data sources in a single database and the future possibility of exposing parts of this information to external users through a web application.

Future integration of external sources could improve the information management of biobank data for postmarket surveillance of medical devices.

Acknowledgements

I want to thank the following people that made this thesis possible:

My supervisor Ankica Babic, for guiding me through the master thesis process and for all your help along the way.

The researchers at the Biomaterials lab at the Haukeland University hospital, for providing valuable feedback along the way and taking time out of their work day to help test the prototype.

Peter Ellison, for always being ready to answer any questions regarding anything from databases to patellas.

Paul Johan Høl, for bringing this project to the department and letting me bother him with interviews and questions.

Fellow students at the Department, for their help in evaluating the prototype.

And finally, Cecilie Bakkedal Langnes, for being a constant source of motivation and encouragement and for listening to me endlessly going on about every small detail of my thesis.

Contents

Abstract	i
Acknowledgements	ii
List of Figures	vi
List of Tables	vi
Code listings	vii
List of abbreviations	vii
1 Introduction	1
2 Research questions	2
3 Theory	2
3.1 Implant technology	2
3.2 Database design	3
3.2.1 Multimedia databases	5
3.3 System development	6
3.3.1 Testing	7
3.4 Methodology	7
3.5 Related work and clinical applications	8
4 Methodology	9
4.1 Design science research	9
4.2 System development	12
4.3 Development methodology	14
4.4 Evaluation	16
5 Development	17
5.1 User needs assessment	17
5.1.1 Interviews	18
5.2 Design patterns	20

5.2.1	Model View ViewModel	21
5.2.2	Mediator	23
5.2.3	Singleton	23
5.2.4	Repository	24
5.3	Development tools	24
5.3.1	Server and software requirements	25
5.4	Database development	26
5.5	Application development	26
5.5.1	Method	26
5.5.2	Architecture	27
5.5.3	Navigation	28
5.5.4	Record entry	29
5.5.5	Record viewing	31
5.5.6	Query builder and exporting	34
5.5.7	Importing registry data	37
5.6	Web application prototype	38
6	Evaluation	40
6.1	User testing	41
6.2	Heuristic evaluation	48
7	Discussion	57
7.1	Methodology	57
7.1.1	System development methodology	57
7.1.2	Research methodology	58
7.2	Design patterns	60
7.3	Prototype	61
7.4	Evaluation methods	62
8	Conclusion	63
8.1	Future development	64
	Appendices	A1
	Appendix A User testing informed consent form	A1

Appendix B	Interview transcript - User 3	A2
Appendix C	Interview transcript - Lab users	A13
Appendix D	User testing task instructions	A16
Appendix E	Nielsen's usability heuristics	A18
References		R1

List of Figures

1	Knee implant statistic	3
2	Projected rise of implant operations	4
3	ANSI-SPARC diagram, adapted from Tschritzis and Klug (1978)	5
4	System development process overview	15
5	Prioritized and estimated user stories	21
6	The MVVM pattern (Microsoft Corporation, 2012)	22
7	Entity relationship diagram	27
8	Application model	28
9	Application main menu	29
10	Create new record screen	30
11	View list of all records screen	32
12	View single record screen	33
13	Blood and tissue sample dialogs	35
14	Query records screen	36
15	Import registry data screen	38
16	Web application record list	39
17	Web application record details	40
18	Task success distribution by task	44
19	Average time on task by task	46
20	Total SUS score	48
21	Average SUS by question	49
22	Heuristic evaluation questionnaire scores	55

List of Tables

2	Overview of the clinical indications, sizes, lengths and responsible parties in clinical trials on knee implants as retrieved from clinical-trials.gov on April 4th 2014	8
3	User testing task overview	42
4	Median and standard deviations in minutes for task times	47
5	Heuristic issue 1	50
6	Heuristic issue 2	51

7	Heuristic issue 3	51
8	Heuristic issue 4	52
9	Heuristic issue 5	52
10	Heuristic issue 6	53
11	Heuristic issue 7	53
12	Heuristic issue 8	54
13	Heuristic issue 9	54

Listings

1	Input validation example	31
2	ViewModel communication	31
3	Direct way of opening dialog window	34
4	Building the query criteria	36
5	Reflecting and writing on record properties	37

List of abbreviations

ASP	Active Server Pages
DBMS	Database Management System
ER	Entity Relationship
GUI	Graphical User Interface
IDE	Integrated Development Environment
MMDBS	Multimedia Database System
MVC	Model View Controller
MVVM	Model View ViewModel
OSS	Open-source software
SUS	System Usability Scale
TDD	Test Driven Development
TFS	Team Foundation Server
WAF	Web Application Framework
WPF	Windows Presentation Framework
XP	Extreme Programming

1 Introduction

The goal of this thesis is to develop a new prototype of a database system and desktop application for the Biomaterials research group at the University of Bergen. The prototype will have to integrate independent data sources and should meet the needs of researchers who normally relies on information from a number of different sources.

It is a design research thesis of implementing a new system in an existing technology environment. The research group is connected to the networks of both the University of Bergen and the Haukeland university hospital. The work that the system will support involves analysing extracted knee implants that have failed and have been removed from the patients. The data generated from these analyses needs to be integrated with data about the patient and the implant itself. Today the researchers get this data from a national registry by requesting a SPSS (IBM, 2014) file from a statistician at the registry. This is then imported to an ad-hoc Microsoft Excel (Microsoft Corporation, 2014c) spreadsheet on a network drive. The current process is cumbersome and takes up a lot of the researchers valuable time. It also has issues with concurrent access. The researchers also examine hip implants, but data concerning those are stored in a networked Microsoft Access (Microsoft Corporation, 2014b) database.

The proposed solution to this problem is developing a database system for integrating the clinical data with the statistical data from the registry. A best case scenario would be the database system making automatic retrievals from the registry services, but that would involve close cooperation with the registry administrators and is outside the scope of this thesis. A desktop application will also be developed in order to let the researchers access the database system. Functionalities will include letting the end users input new data and get presentation of existing integrated data. Elements from the Scrum development methodology that can be adapted for use by a single developer will guide the development process.

In the long term, the database would ideally be retrieving data from different external data sources, for example similar databases that are developed in order to support international cooperation in this field of research. To demonstrate the feasibility of enabling external access to the system, a web based prototype will be developed in addition to the desktop application.

In summary, this thesis will attempt to collect data from multiple clinical data sources, integrate them in a database system and finally present the available data to the end users through a desktop application. A user evaluation combined with heuristic evaluation is planned to assess the functionality of the finished system.

2 Research questions

The research in this thesis focuses primarily on three separate questions.

- How can design science research methods be used to develop a new computer based system for the domain of knee implant failure?
- What is the potential impact of transferring the current state of information processing to a database system on the research, information management, and data sharing?
- How can elements from agile development methodologies be adapted for use in a single student developer setting?

As an additional research question, the thesis will also explore the advantages and disadvantages of using established design patterns in order to encourage code reuse and maintainability in a student development project.

3 Theory

This section will draw on literature mainly from the fields of database design and system development. I will also introduce the clinical area of implant technology business domain in which this thesis seeks to address an existing problem. Literature concerning development methodologies, and in particular agile methodologies, concludes the chapter.

3.1 Implant technology

The planned prototype will get its patient- and implant-data from a national register. The researchers focus on implants made of titanium alloys, materials

which shows superior bio compatibility and wear resistance compared to traditional materials (Long & Rack, 1998). Even though titanium implants have been widely used for decades and help the quality of life of many patients, these implants can still fail, which results in painful invasive surgery for the patient and increased costs for the health services (Noort, 1987). Statistics from the registry show that the number of patients getting both hip and knee implants have risen considerably in the recent years (Haukeland University Hospital, 2013), see Figure 1.

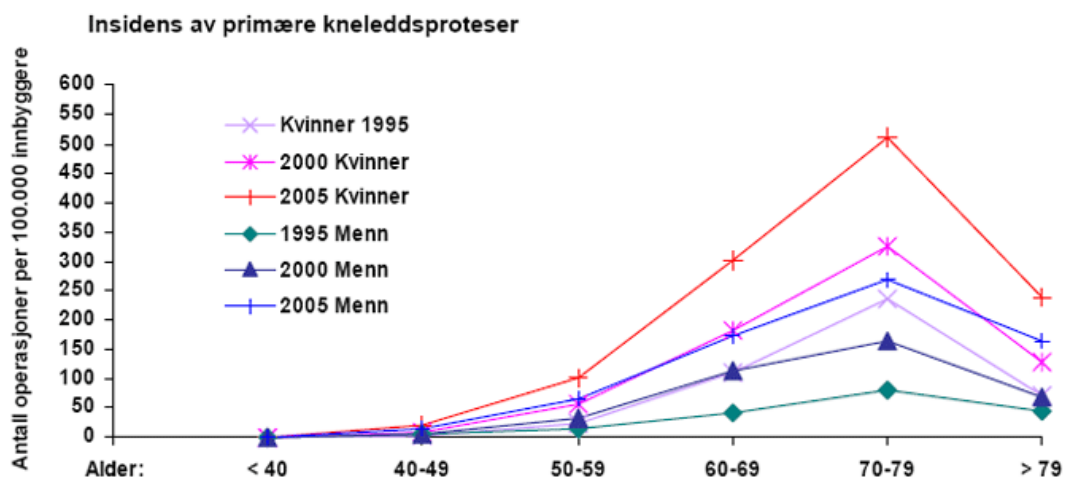


Figure 1: Knee implant statistic

Due to the general population of Norway living longer on average, and the relative large size of the post-war generation (Ræ der, 2012), the amount of patients needing joint implants is projected to rise dramatically in the years to come. Statens Helsetilsyn estimates that the number of hip replacements done annually will rise by 40% for women and 70% for men by the year 2030 (StatensHelsetilsyn, 1999), see Figure 2. This highlights the need to better understand how and why implants are failing. In order to do this, clinical researchers need suitable IT tools.

3.2 Database design

Using the ANSI-SPARC (Tsichritzis & Klug, 1978) conceptual model (see Figure 3) as a tool to divide the database design into three separate layers, it is easier to get input from the end users when working within the conceptual and presentation

i) Hoftepoteser (insidens)

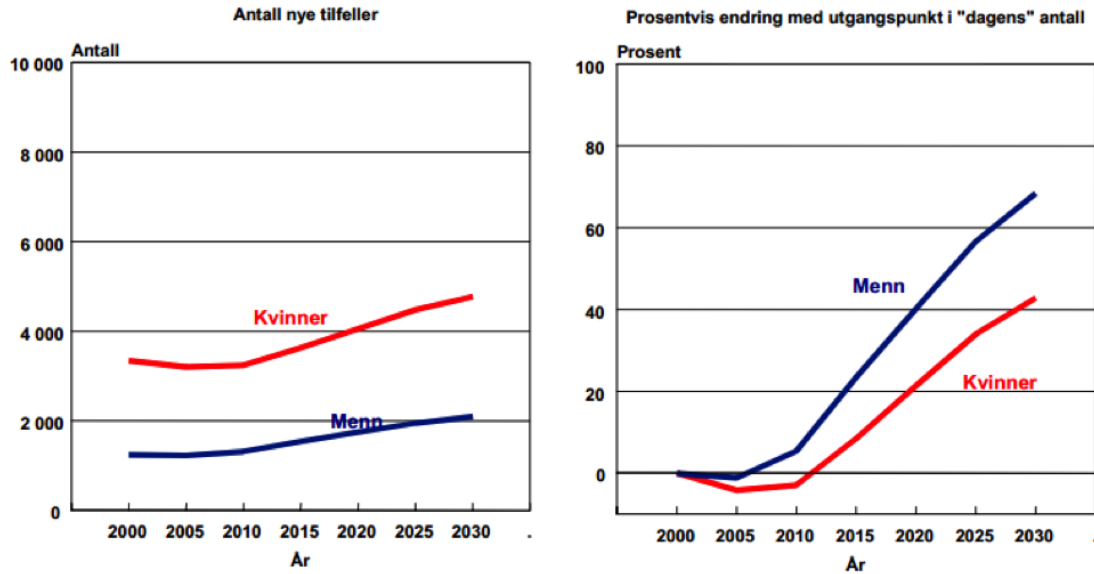


Figure 2: Projected rise of implant operations

layers without having to go into detail about implementation issues. This separation also makes it possible to make changes at the presentation level without necessarily changing the conceptual and internal levels.

Conceptual modeling was done by means of the Entity-Relationship (ER) modeling language (Chen, 1976). This enabled a non-technical view of the database structure, useful for communicating with the end users. Later in the process, it can also be automatically converted to the table structure found in relational databases (Coronel, Morris, & Rob, 2012). Extending the ER model with the additional concepts offered by the Extended Entity-Relationship (EER) model can also provide the database models with more precise semantic content such as sub classes and superclasses, for example a “knee implant” being a subclass of the supertype “implant” (Navathe & Elmasri, 2010).

In the modeling step, it is crucial that the developers and end users see the data in the same way, thus communication should be frequent and clear at this stage (Coronel et al., 2012). This was, together with the user interface design and evaluation stages, the point in the project where most of the communication with the end-users occurred.

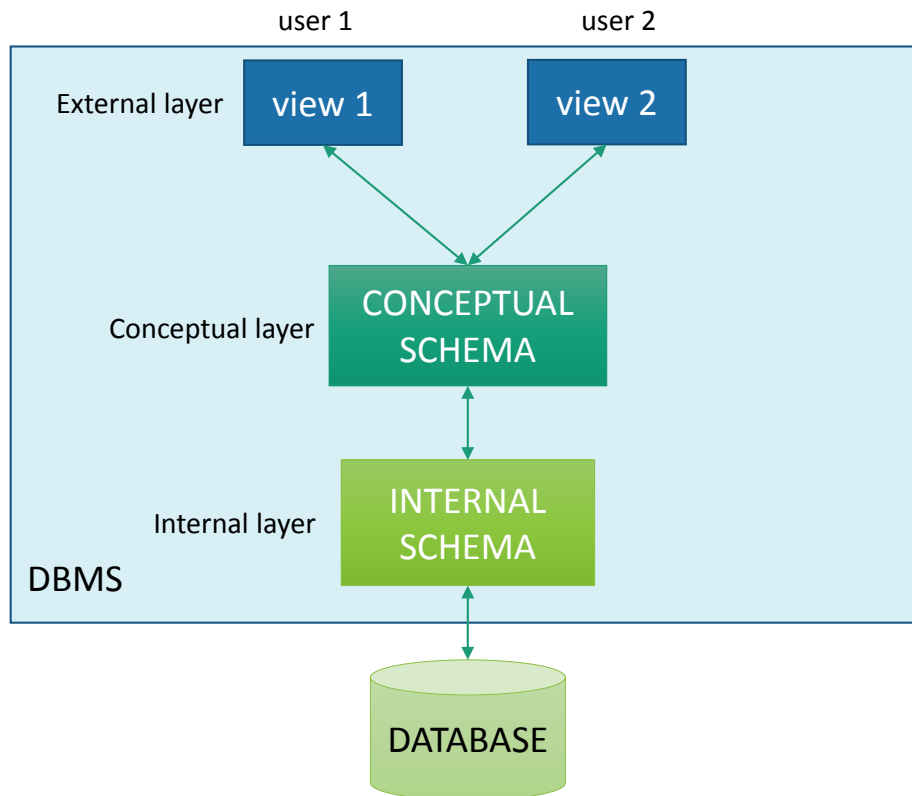


Figure 3: ANSI-SPARC diagram, adapted from Tsichritzis and Klug (1978)

3.2.1 Multimedia databases

Initially, since much of the clinical data generated by the researchers is in the form of images, the prototype was thought to have to use a Multimedia Database Management System (MMDBS) to support multimedia data types (Kosch & Döller, 2005). However, it was a request that the multimedia data were to be stored on a network drive separate from the database because these images were also being used by other users and other applications. Doing content based image queries (Müller & Deserno, 2011) or any sort of automatic image analysis is outside the scope of this thesis. However, in the long term, allowing tighter integration of multimedia data might be an option and as such should be taken into consideration when designing the prototype.

3.3 System development

The main prototype was initially supposed to be a local web application and a basic web prototype was also developed as a proof of concept.

The term web application can mean different things to different people, but Conallen (1999) defines it simply as being an application that delivers its front end mainly via a web system. This broad definition covers most of the more specific definitions available. An important point to make here is that a web application is not necessarily connected to the internet, it can also exist entirely on a web server situated inside a local intranet. Operating within an already robust intranet simplifies some of the security concerns associated with internet-enabled web applications.

There exist a multitude of different frameworks for developing web applications. Shan and Hua (2006) define a web application framework (WAF) as a reusable, skeletal, semi-complete modular platform that can be specialized to produce custom web applications. A WAF aims to speed up development time by providing some building blocks in order to standardize common-performed development tasks. They also serve to provide some much need structure to the sometimes ad-hoc programming of web applications (Doyle & Lopes, 2008). Most WAFs utilize a Model-View-Controller (MVC) (Krasner & Pope, 1988) structure in order to separate application logic, data and presentation (Jazayeri, 2007). This ensures a loose coupling between the different parts of the application. The Microsoft ASP MVC WAF was used to develop the web application prototype since it has synergies with the core application and allowed both data storage and access logic to be completely reusable.

To select the framework that is most suited for a particular project, some practitioners recommend doing 1-week spikes of different frameworks on the same toy project in order to get a feel for them and compare efficiency (Raible, 2012). However, in this case, the web application prototype was started after having already implemented the desktop application using Microsoft technologies. Because of this, the ASP MVC approach was the most reasonable option as it allowed both using the same software tools and also much of the existing code used for the desktop application.

Cimino, Socratous, and Clayton (1995) summarize why web applications can

work well in a clinical setting. They emphasize the ability to present information to multi-platform clients, use of standardized protocols and the ease of which other internal or external data-sources can be integrated into the application. This early research into web based clinical software has proven to be relevant even as the internet technology has gone through massive changes since these findings were first reported.

3.3.1 Testing

Testing in this context refers to the concept of executing certain parts of the program code in order to verify that it meets certain requirements. This sort of technical testing should not be confused with user testing where the actual users of the software interact directly with the system to assess its performance and usability. Automated testing should be a key part of the system development process. Extensive and continuous testing of the software is a way to validate that the program performs as expected (Olan, 2003), and is one of the most important practices in agile software development (Abrahamsson & Warsta, 2003). Most web application frameworks have built in testing functionality to make it easy for the developers to integrate a good automated testing policy. Test-driven development (TDD) was considered as an option, but lack of experience in this area of development resulted in it not being used for the prototype implementation.

3.4 Methodology

Most methodologies, and especially agile methodologies assume that developers will be working in a team since developers working completely isolated is extremely rare in the business environment that these methodologies are designed for. However, efforts have been made to adapt agile methods such as Extreme Programming (XP) (K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mallor, Ken Shwaber, 1999) for situations with a single developer (Agarwal & Umphress, 2008). Naturally practices such as pair programming is not possible when working in a one-man team, but Agarwal and Umphress (2008) refine most of the other aspects of XP to fit in such a situation. There have also been attempts to adapt the more plan-driven traditional Personal Software

Process (PSP) (Humphrey, 1996) with agile methodologies (Shen, Rong, & Shao, 2013).

3.5 Related work and clinical applications

In addition to administrative reasons there are many good reasons to create and maintain registries of patients that have received knee implants. One is the research that explores the nature of the long time survival and side effects.

The other is to explore a number of devices and explore their performance, duration, and even their own evolution over time. The longer the length of observation, the better insight into patient outcomes such as quality of life, mobility, need for follow ups and interventions.

Knee implant research is an active field with many ongoing clinical studies. Long term trials are required to evaluate the effects of the implant over time. Table 2 gives an overview of the top 5 ranked current clinical trials from ClinicalTrials.gov (U.S. National Institutes of Health, 2013)

Indication	Patients	Study length	Reponsible party
Osteoarthritis of the Knee Joint	150	4 years	University Hospital, Ghent
Amputation	10	1 year	Providence VA Medical Center
Total Knee Arthroplasties	100	2 years	ConforMIS, Inc.
Total Knee Arthroplasties	50	1 year	ConforMIS, Inc.
Arthroplasty, Replacement, Knee	67	5 years	Radboud University

Table 2: Overview of the clinical indications, sizes, lengths and responsible parties in clinical trials on knee implants as retrieved from clinicaltrials.gov on April 4th 2014

The longevity and scale of large clinical studies such as these will generate huge amounts of data that could potentially be useful for researchers both for the responsible parties and for other external institutions. Suitable software solutions for managing this information is therefore crucial.

4 Methodology

The following section will provide the methodological context for this thesis. An overview of design science research and system development as a research methodology is presented, along with a discussion of how this thesis relates to these approaches.

4.1 Design science research

Hevner, March, Park, and Ram (2004) divide information systems research into two distinct paradigms - behavioral science and design science. The former has roots in the natural sciences and concerns itself with explaining phenomena that explain the development and usage of information systems (March & Smith, 1995). Design science on the other hand has its roots in engineering and aims to “extend the boundaries of human and organizational capabilities by creating new and innovative artifacts” (Hevner et al., 2004). An important point to make is that the use of the artifacts produced by design science can lead to phenomena that can in turn be studied by behavioral science (March & Smith, 1995).

This thesis follows the design science research paradigm.

Hevner et al. (2004) provide a set of guidelines for conducting out and evaluating design science research:

- Guideline 1: Design as an Artifact
 - According to Hevner et al. (2004) the desired result of design science research in IS is an IT artifact that addresses a certain organizational challenge. Definitions of the term IT artifact differ between researchers. Orlikowski and Iacono (2001) emphasize the artifact as an instantiation of software and hardware, and also takes into account the developers, the organization and the users themselves involved in the interaction with the artifact. Hevner et al. (2004) has a different definition of the term - they include the models and methods used in developing the system. They include the USE of the system, but not the actual organization and people using it.

This thesis aims to deliver a working prototype as its final result. As it is being developed in cooperation with external end users, the organi-

zational and people factors will be more important than in an isolated research environment. The methods used in developing the system will also be part of the evaluation. If the national registry can expose its data through a service at a later point, the system can be seen as automating a process which is currently being done manually.

- Guideline 2: Problem Relevance

- The prototype resulting from this thesis seeks to resolve a real problem - researchers are spending too much of their time managing inefficient information systems. Hevner et al. (2004) define a problem as the differences between a goal state and the current state of the system. The goal state for this project can be loosely defined as a situation where the researchers spend as little time as possible on performing administrative IT tasks while still getting the information they need to perform their research. Hevner et al. (2004) also argue that in order to overcome challenges of technology acceptance among end users, organization-based artifacts and and people-based artifacts such as social systems and training need to be addressed in addition to the IT system artifact itself. An examination of current work practices of the researchers in the early phases of the project will be crucial to the prototype's success. Sufficient demonstration and training of the system will also be needed after the completion of the prototype system.

- Guideline 3: Design Evaluation

- Evaluation of the prototype will be done in cooperation with the users. The evaluation should also cover how well the system is integrated in the general IT infrastructure of the organization (Hevner et al., 2004). Following an agile continuous delivery approach in the development of the application, evaluation can be done incrementally as key features are implemented. Evaluation will be covered in greater detail in section 4.4.

- Guideline 4: Research Contribution
 - A design science research project must make contribution in at least one of three areas - the design artifact, design construction knowledge or design evaluation knowledge (Hevner et al., 2004). The main contribution of this project will be the developed prototype system. It aims to address the currently unsolved problem of integrating clinical data with statistical data from the external data source. Its secondary contribution is exploring how the use of well established design patterns can enable code reuse in a student development project, both for each developer and between developers.

- Guideline 5: Research Rigor
 - Research rigor concerns the way in which the research is conducted (Hevner et al., 2004). An over-emphasis on mathematical formalism often used in design science research can weaken the relevance of the research due to abstracting away important parts of the issue (Hevner et al., 2004). As this project relies heavily on human-machine problem solving, no mathematical methods of evaluation will be performed on the resulting prototype. Instead, the users will test the system according to predefined criteria. As Hevner et al. (2004) point out, the aim is to determine how well an artifact works, not theorize on the reasons why it works. This project has the advantage of having “real” users to test the prototype, avoiding some of the possible challenges with selecting suitable subject groups for testing.

- Guideline 6: Design as a Search Process
 - In the sixth guideline, Hevner et al. (2004) describe the research process as an iterative search process to discover the best solution to a specific problem. The authors state that, “Means are the set of actions and resources available to construct a solution. Ends represent goals and constraints on the solution. Laws are uncontrollable forces in the environment.” (Hevner et al., 2004).

- Guideline 7: Communication of Research
 - The final guideline states that design science research has to be presented differently for technology-oriented and management-oriented audiences. Technology-oriented presentations will need to describe the implementation in enough detail to be constructed, while management-oriented presentations need to tell the audience if the artifact is suited to spending resources on in their organizational context (Hevner et al., 2004). This thesis will have to deal with both communication techniques, focusing more on the technical-oriented way in the thesis and in a more management-oriented way to the researchers using the system.

4.2 System development

Research can be classified along several different axes. Nunamaker and Chen (1990) define IS research to be applied science and belonging to the engineering rather than scientific approach, being developmental rather than evaluative and formulative rather than verificational.

This thesis will mainly use system development as its research methodology, however, as Nunamaker and Chen (1990) point out, the systems development research methodology can be combined with various other approaches such as field studies. Preliminary phases of the thesis will use elements of case study methodology in order to determine how the current system is being used by the researchers and to explore features that will be demanded of the new system. The final evaluation phase will also involve observation and interviews with the users interacting with the finished prototype.

Nunamaker and Chen (1990) propose a systems building process involving the following phases:

- Construct a conceptual framework
 - This phase involves defining a relevant research question and early investigation of the system requirements and functionalities. The first part of this phase will be covered by the work done in the INFOMEVI300 course. The latter part will be done in cooperation with the users during the initial period of the project work. Justification of the development

of a new system comes from the inefficiency of the current system as perceived by the users.

- Develop a system architecture
 - The second phase covers solidifying the functional requirements of the system. This gives a basis for evaluation in the later stages. Identifying any environmental architectural constraints is also crucial at this point. Possible constraints the system will have to work within includes the current IT infrastructure of the users and their network, cooperation with the national registry and the work practices of the researchers.
- Analyze and design the system
 - After acquiring a sufficient understanding of the system requirements and the problem domain, data structures and the system database will be designed. By replacing a current database instead of designing one completely from scratch, some of the answers to the issues faced in this phase can hopefully be extracted from the current system. Initial design of the application is another task at this stage.
- Build the system
 - When the initial design of the database and application is completed, the prototype will be implemented. During this process, the problem that the system is trying to overcome will be better understood.
- Observe and evaluate the system
 - The final phase will include field studies observing the users interacting with the prototype. Both the system itself and the process of developing a system in a single developer setting using agile methods will be the units of analysis for the evaluation. This is covered in more detail in a later section. See section 4.4 - Evaluation.

Although Nunamaker and Chen (1990) describe stages more in line with a rigid traditional “waterfall” system development methodology, the various research concerns at the different stages will be applied to the corresponding points in an agile development process.

4.3 Development methodology

Choosing an appropriate methodology for development should guide the development work. Kanban is widely used by information science master projects at the Department. Many former students cite its ease of use as one of the reasons why they chose to use it. Kanban has few rules and as such is a very lightweight methodology (Reese & Iberle, 2011). This might very well resemble real-world development situations in the industry, where many loosely defined agile methods are being used. The lack of structure in this approach can be seen as an advantage in the sense that the risk of getting too much time tied up in administrative tasks is reduced. However, this also means that there are no predefined practices that demands particular deliveries at different stages.

For this thesis the choice was between Kanban and a variation of Extreme Programming (XP) and Scrum practices and procedures customized for single developers. All are considered highly agile methods, but XP can be seen as the more structured approach.

In this thesis, no established development methodology is followed one hundred percent. Instead, some of the elements from the Scrum methodology that can be adapted for use by a single developer will be applied. This includes expressing the system requirements in terms of user stories and dividing up the phases of the development into sprints.

A user story is a representation of a system functionality expressed from the users perspective. Many agile practitioners have adopted the following template suggested by Cohn (2004): “*As a <type of user>, I want <some goal>, so that <some reason>*”. An example of this might be: “As a system administrator, I want to automate backup processes, so that I am not required to manage them manually”. The main advantage of user stories is that they are written in an every day language free from technical details, which makes them suitable to discuss with any users in any field. The concept was first introduced by the XP movement, but has also been incorporated by other agile methods such as Scrum.

The product backlog is a Scrum concept that utilizes the user story. It is a list of all the user stories for a given product. They are then moved from the product backlog to the sprint backlog, which is a list of the user stories included in a specific sprint. In Scrum, a sprint is a timeboxed (fixed duration) iteration.

A sprint is usually of one month or less in length, and the product of a sprint should be a potentially releasable product (Schwaber & Sutherland, 2013). After a sprint is finished, a sprint retrospective and a sprint review should take place. A sprint retrospective involves the team reflecting on what went well in the previous sprint, what did not work and what could be improved. In a sprint review, the team presents the releasable product from the sprint to the product owner who then gives his verdict of whether or not the included user stories for that sprint should be considered “done”. If a user story is not done in the eyes of the product owner, that user story will go in to the next sprint.

Figure 4 summarizes the general system development outline. It presents the different information sources that went into developing an outline of the prototype system. This then leads to the three development phases shown in grey. The figure presents a high level overview of the entire project and does not show any specifics about the development process.

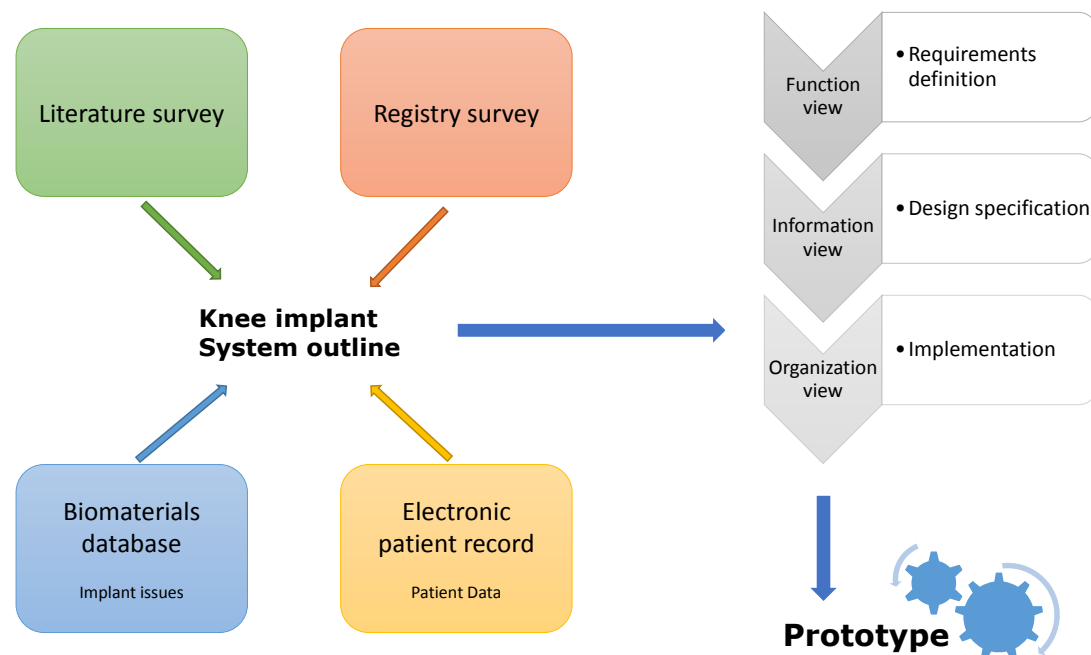


Figure 4: System development process overview

4.4 Evaluation

In this context, evaluation is referred to as the final formal testing of the system. A form of evaluation will also be occurring continuously as the prototype is being developed. Evaluation of the prototype will be done by means of user interviews and observation of the users interacting with the system. The observation will preferably be performed in a non-participatory manner. It will be done overtly, with a full explanation to the users of what is being tested. Data recording can be both informal note taking and technology based, such as screen capturing and video recording. The testing should be spread out over at least two sessions, in order to let the users get familiar with the system. Testing initial reactions might be beneficial in evaluating the aesthetics of the user interface, but in terms of measuring ease of use and effectiveness, testing users that have some experience with the system would be better.

Heuristic evaluation was used as a secondary means of evaluation in order to gain feedback from more technically proficient participants. Heuristic evaluation is done by simply looking at the interface and making a subjective assessment of what is good and what is bad about it. Nielsen and Molich (1990) emphasize that this kind of evaluation is ideally done by following certain rules. They critique earlier approaches that utilize a very high number of rules and present their own set of heuristics limited to nine rules. These heuristics were later refined and expanded by Nielsen (1995) into a set of ten rules that are still frequently used by evaluators:

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use

8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

Full explanations of these heuristics as described by Nielsen can be found in Appendix E.

5 Development

5.1 User needs assessment

The initial work on user needs assessment began months before the actual start of the thesis work, at a meeting with the users. At this point the limitations of the current solution and high level goals of the potential prototype were discussed. While the technical details of the prototype were still not clear, the high level architecture plans were clear and the strategy was to implement as a local web application. The reason for this approach was that it might later be further developed into an internet-enabled web application offering access to external parties such as other researchers or clinicians.

When the actual project work started a few months later, it became clear that the plans for external access were not in the immediate future.

This fact, coupled with security concerns, led to taking different approach to the prototype architecture. The prototype would now be built as a desktop application. The next logical step was to determine the target operating system, or if needed, which cross-platform solution to choose. This was quickly narrowed down to either a cross-platform Java solution or a Microsoft Windows only .NET solution. .NET is a Microsoft development framework for developing applications for the Windows operating system (Microsoft Corporation, 2014d).

After examining pros and cons for both alternatives, a one week “spike” was performed with each alternative. A spike is a concept that stems from the Extreme Programming development methodology (Jeffries, 2009). It is a short experimental prototyping process that is meant to explore possible solutions to a problem. It is a means of minimizing risk when dealing with unfamiliar technologies.

The main advantage for a Java solution was that the University of Bergen, especially the Department of Information Science and Media Studies, primarily uses Java as the language of choice in programming courses. This means that the author and potential future student developers would already have some level of Java proficiency. Another advantage was that a Java solution could run on more than just the Windows operating system, for instance Mac OS and Linux. The disadvantages were a more fragmented development environment, using specialized software for many tasks of the development process, such as database design, coding and visual design. The one week spike also showed this alternative to be more time-consuming.

The Windows-based .NET solution would involve using the C# programming language coupled with the Windows Presentation Foundation (WPF) for graphical user interface (GUI) work. WPF is a GUI system for developing Windows applications (Microsoft Corporation, 2014a). It is a part of the .NET platform.

Although C# is not commonly taught in Norwegian university courses, it has a syntax that closely resembles that of Java, and many of the language concepts are also similar. This makes a switch to C# for a student well versed in Java a relatively painless transition. The development environment for the .NET alternative is the opposite of the Java alternative. Using Microsoft Visual Studio, all phases of the development could be done using the same software. Implementing the spike with .NET was substantially faster than with Java.

One of the main advantages with Java proved to be of very limited importance. As all the University of Bergen computers at the Biomaterials lab were running Windows operating systems, the need for cross-platform support in the foreseeable future was not there. One exception is that one of the researchers has a Mac OS computer, but he is already using virtualization software to run other Windows applications.

Based on these pros and cons, the decision to use .NET as the architectural platform was taken with the admin user.

5.1.1 Interviews

Further examination involved semi structured interviews with the users individually. Oates (2005) defines a semi-structured as having predefined questions, but

the order can be changed depending on the situation. Follow-up questions that are not predefined can also be asked. This was the approach taken for these sessions. The aim was to allow the researchers to express their thought about their information management freely, but still within the bounds of the predefined themes. The following questions served as the guidelines for the interviews, but were expanded upon, altered or omitted depending on the user role of the interviewee.

- What is your age, educational background and position?
- What are your daily job tasks?
- How do you use the hip database in your work?
- How often do you use the system?
- How well do you feel the system is supporting your daily work?
- Do you see any areas that have a potential for improvement in today's system? Which ones?
- Are there any concrete functions you feel are missing in today's system? Which ones?
- Does the system for knee implants have any limitations compared to the system for hip implants?

These interviews explored how the users interact with the current system in their daily work. The interview subjects were users who filled different roles in the organization. The unique user roles are important to study in detail as they might have different interactions with the system and therefore have different needs (Cohn, 2004). The users gave their informed consent that the interviews would be recorded, transcribed and possibly used in the thesis. They were informed that all personal names and details would be anonymized. In order to make the interviewees feel comfortable, the venues for the interviews were their everyday working environments such as labs and offices. This has also satisfied the recommendation from Oates (2005) that the venue should be private and reasonably quiet. Transcripts of the user interviews can be found in Appendix B and Appendix C. The

interview sessions concluded with an observation of the users showing examples of their interactions with the system from their workstations.

The interviews were very useful at this stage of the project, as they served to increase the understanding of current working practices and processes. Some limitations and ideas for solutions that were not previously discussed also surfaced during these interviews.

It was a goal to incorporate the user story concept as the way to document system requirements.

The users did not have experience of working with this way of expressing functionality, so some effort was put into explaining the advantages and disadvantages of this approach.

The admin user was chosen as the product owner and had the main responsibility for defining user stories.

His stories were refined and expanded upon based on the information gathered in the user interviews. The product owner would not be in Bergen during the development of the prototype, so a distributed approach to the agile process was needed. User stories were recorded, prioritized and moved between states in the Trello.com web application (Fog Creek Software, 2014).

Figure 5 shows the initial requirements for the prototype expressed as user stories. It defines three distinct user roles: Admin - the administrative user, G user - general user, L1 user - lab user. User stories are prefixed by the user role that the story is relevant for. The user stories were prioritized by sorting them in a descending order on Trello, where the first item is the most important one. This prioritization was done by the product owner. The stories were also estimated in terms of expected development complexity and time usage. This was done by means of the coloured labels also shown in Figure 5 where a score of 8 is the most complex story and a score of 1 is of the least complexity.

5.2 Design patterns

A goal when developing the prototype was to utilize established design patterns whenever applicable and relevant. Design patterns aim to generalize and structure recurring important designs that appear frequently in a object-oriented system. They make it easier to reuse successful designs (Gamma, Helm, Johnson, &

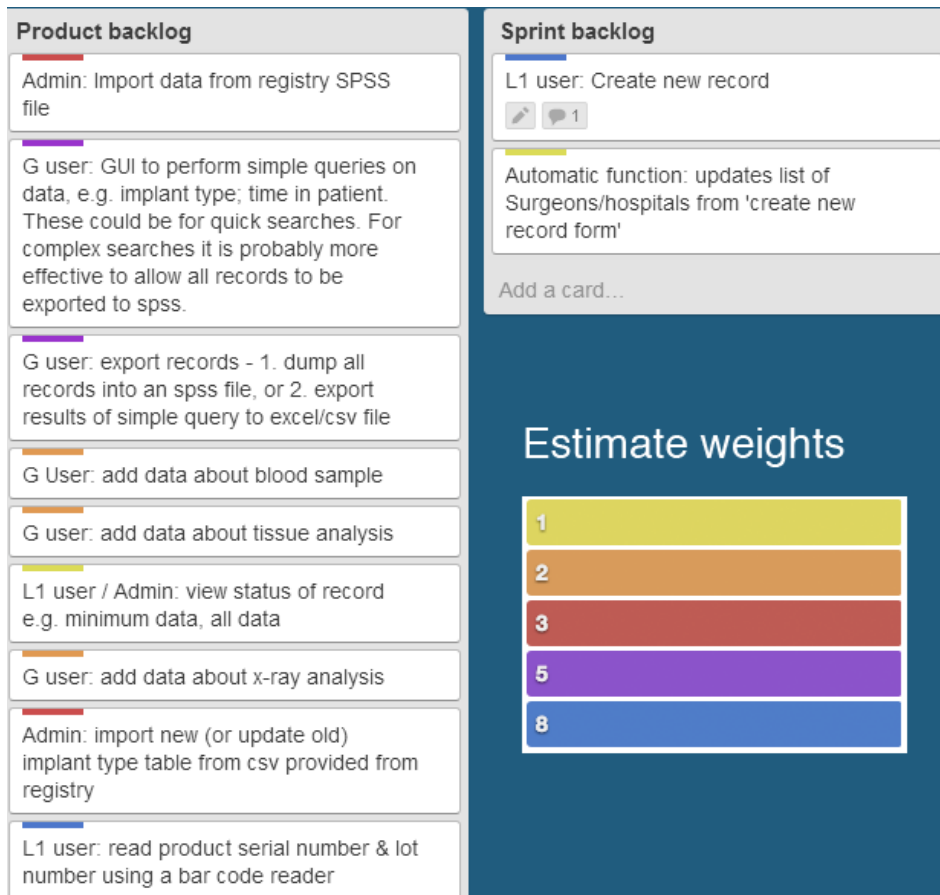


Figure 5: Prioritized and estimated user stories

Vlissides, 1994). This is especially important in a project such as this that will potentially be handed off to other developers.

5.2.1 Model View ViewModel

Model View ViewModel (MVVM) is a high level architectural design pattern first described by Gossman (2005). It aims to help separate the many different concerns when developing a GUI, such as the logic behind the application, the visual design, connectivity, multi-threading, security and input validation (Smith, 2009). This separation allows for a greater re-use of components and also makes it easier to work on each layer of the application independently of the others. That way a visual designer can do his design work with almost no knowledge of the underlying

code.

The MVVM pattern breaks an application into three distinct parts - the model, the view and the viewmodel (Gossman, 2005). The data that the application displays is the *model*. In this thesis, this part only represents data and does not contain any business logic at all. However, some MVVM users do not use this approach of a “dumb” model and instead allow the model to handle the business logic. The *view* consists of the actual user interface that the user sees and interacts with directly. The *viewmodel* exposes the data from the model to the view. The interaction with the model objects, and in this project the business logic, is handled by the viewmodel, never the view (Smith, 2009).

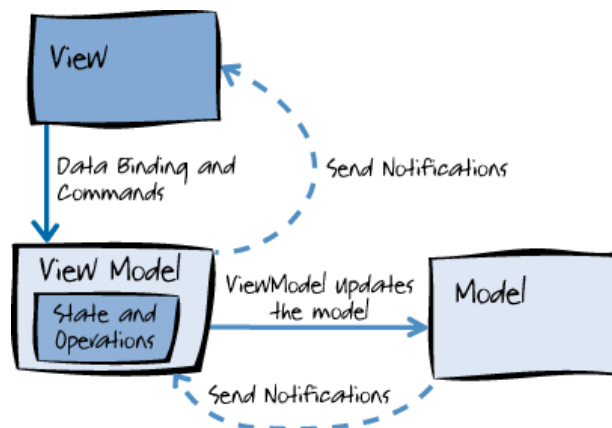


Figure 6: The MVVM pattern (Microsoft Corporation, 2012)

MVVM achieves its separation of concerns by using loose couplings between these three parts of the application. The view is only aware of the viewmodel, not the model itself. The viewmodel is aware of the model, but not the view. Instead it lets the views bind to properties of the viewmodel. This enables multiple views to use the same viewmodel. The model is completely oblivious to both the view and the viewmodel (Smith, 2009). These relationships are visualized in Figure 6. One of the goals of choosing this approach for this project was to allow more design-focused potential future developers to focus solely on the user interface of the application without having intimate knowledge of the code behind.

Another advantage of this loose coupling is that automated testing becomes easier. Testing that involves automating user interface (UI) operations is time consuming both to create and run. Using MVVM eliminates the need to involve the UI

in testing the application logic, the tests can interact solely with the viewmodels.

5.2.2 Mediator

In order to further ensure loose couplings in the application, another design pattern is necessary. Ideally the different viewmodels should be independent and even not be aware of each other. However they still need to communicate and pass information between them, such as when navigating to a new view/viewmodel or opening a specific element in a list in one screen and then passing that element to the next screen that will be displayed.

To solve this challenge without the viewmodels having explicit knowledge of each other, the mediator pattern (Gamma et al., 1994) is used. The mediator pattern solves this by implementing a separate class that is responsible for all communication back and forth between the different participants.

By using the MVVM Light Toolkit (Bugnion, 2013), the Messenger class, an implementation of the mediator pattern specifically tailored for WPF and MVVM is available. Using the Messenger class each viewmodel has the opportunity to register for different types of messages that the Messenger broadcasts to all subscribers. The viewmodels can also send these messages to the Messenger, which then relays them to all other participants that are registered for that type of message. This approach relies on a single Messenger to handle all the communication between viewmodels in the application, which leads the following design pattern.

5.2.3 Singleton

The Singleton pattern is among the simpler design patterns commonly used. It handles the responsibility of creating a single object of a class and ensures that only one object is ever created (Gamma et al., 1994). It also provides a global access point to this object for other parts of the application, usually in the form of a static method. In this project the pattern is used both for the Messenger object described in the preceding section, as well as for a data access object that follows a repository pattern.

5.2.4 Repository

The repository pattern enables a developer to encapsulate the data access layer of an application in a separate class (Khan, 2012). This hides the details of the database interaction logic from the other parts of the application. An advantage of this modularity is that changing parts of this interaction will only have to be done in one place - the repository.

Another advantage to the repository pattern is that it greatly simplifies automated testing. By creating a repository object with a fake implementation supplying only test data, unit tests can be run without interacting with an actual database. This brings the amount of time required to run the tests down by a significant amount. In the test project of the prototype solution, this exact strategy is used.

5.3 Development tools

The choice of using the .NET architecture alternative greatly influenced the development tools used for the project. Choosing the Java based alternative would enable the project to use primarily open-source tools. The use of open-source software (OSS) can lead to both benefits and drawbacks, both from a technical and from a business perspective (Morgan & Finnegan, 2007). Some of the benefits Morgan and Finnegan (2007) mention are a lower cost of licensing and an increased focus on compatibility. OSS can also benefit from frequent bug fixes and upgrades if the project is active.

On the other hand, customizing OSS can be costly and projects can die off as there are no commercial backers that have guaranteed to support the software. Using proprietary software usually also means that support is readily available from the vendors.

Choosing tools by a major commercial vendor such as Microsoft means that there is a large community around its products and help and documentation will be available also in the future. It also means that the availability of developers with the requires skills will be high. Additionally, this project uses free editions of the different development software, so licensing is not an issue. These are more basic versions of paid versions with additional functionality and support. The only exception is Resharper (JetBrains, 2014) which is not free, but this tool it not an

absolute requisite for the development of this project. Resharper is a code analysis and refactoring tool that was used to improve the quality of the code.

The tools used for this project consisted of mainly those provided by Microsoft:

- Database:
 - Microsoft SQL Express database server (Microsoft Corporation, 2014e)
 - Microsoft SQL Management Studio (Microsoft Corporation, 2014f) for ER modeling tools and administration
- Application development:
 - Microsoft Visual Studio 2013 (Microsoft Corporation, 2014h) as the integrated development environment (IDE)
 - Resharper for improving code quality

Team Foundation Server (TFS) (Microsoft Corporation, 2014g) was used for version control in the project. Git (Git-scm.com, 2014) was also considered, but at the time Visual Studio provided a more integrated TFS support than it did for Git. This has changed since the project work started and Visual Studio now also has integrated Git support.

5.3.1 Server and software requirements

The requirements for the server hosting the database and the web application server were fairly modest in terms of processing power and memory. The initial request to the IT department stated that disk space was a concern. However, this changed after learning that the database would not store the actual medical images themselves, but only references to their location on a network drive. The IT department set up a Microsoft Server 2012 server for the project use. The server is within the university intranet and does not have internet access. Microsoft SQL Server Express database server was installed. The Express version of SQL Server is a free version, but has a 10 gigabyte size limit. If the limit is deemed too low at some point in the future, alternatives include upgrading to a paid version with no size limits, or migrating to a free open source alternative such as MySQL (Oracle Corporation, 2014). However, ever reaching this limit with the current database structure and information needs is highly unlikely.

5.4 Database development

The structure of the database had some constraints as the data coming from the registry should be stored in the same form as it was imported in. The data generated at the Haukeland research group should also be stored in a similar way as the existing hip Access database. However, the database model used for that database had some challenges, such as not being able to store information about previous revisions for a patient. This issue, along with its solution, is explained in more detail in Chapter 7.3.

The database model for the prototype consists of customized data structures from the hip database, structures that are aligned with the data structure of the registry information and also some novel elements. These elements include storing a copy of the registry data for each record so that this information is kept and associated with a particular record. This removes the need to keep multiple SPSS files and going through them manually to find a certain registry data value for an old record.

Figure 7 shows a partial view of the ER diagram for the database. It shows all the tables, but not all of the properties for all of them. This is due to space concerns, as some of the tables are too complex to display in their entirety here. As an example, the Registry table has 862 properties.

5.5 Application development

In this chapter all of the main functions of the application will be presented in some detail.

5.5.1 Method

Two main concepts from agile methodologies were put into use during the development phase. The different iterations of the prototype were developed in sprints. These were timeboxed to a length of 2 weeks. The sprint retrospectives usually require a team discussion, so no formal retrospectives were conducted. Instead, an effort was made to reflect on the execution of the previous sprint, but this is not as valuable as a discussion where one's ideas can get challenged by others. Sprint reviews were held with the product owner for most of the sprints. Some of them

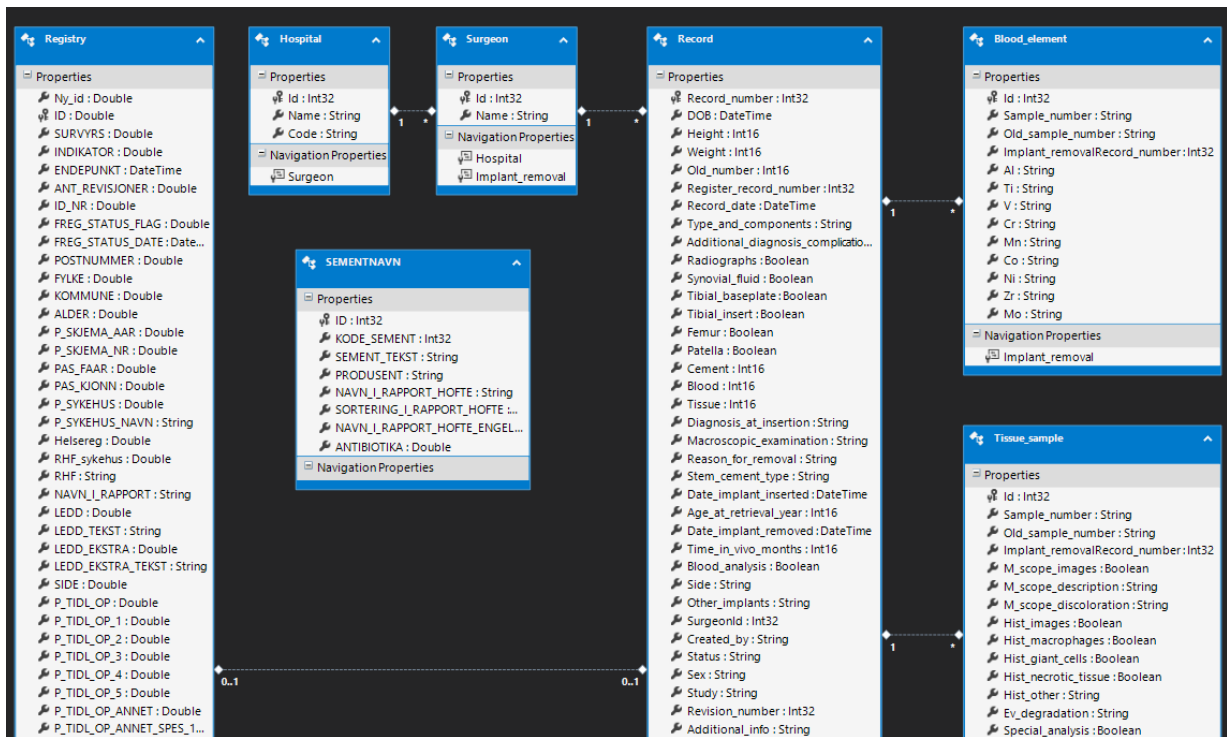


Figure 7: Entity relationship diagram

bled into the next sprint, something which is not ideal, but was caused by practical concerns. Following the sprint approach was not without some difficulties. A few of the sprints were extended in time in order to align with the schedules of both the developer and the product owner. In addition to this issue, one sprint was extended because a releasable product could not be produced in time because of technical development problems.

The user stories defined in the user needs assessment were put into a sprint backlog for each iteration. These were pulled from the product backlog that was prioritized by importance by the product owner. Most of these coincided with a specific screen in the application, so most of the user stories resulted in a visible improvement of the prototype.

5.5.2 Architecture

Figure 8 shows an overview of the different parts of the extended system and the information flow between its elements. Elements in green are parts of the

developed prototype for this thesis. Note that this model also shows potential external users interacting with the web application. The admin user will have to manually import records to the database through the application unless this process can be automated at some point in the future. The model also shows the role of the Repository class discussed in the previous chapter.

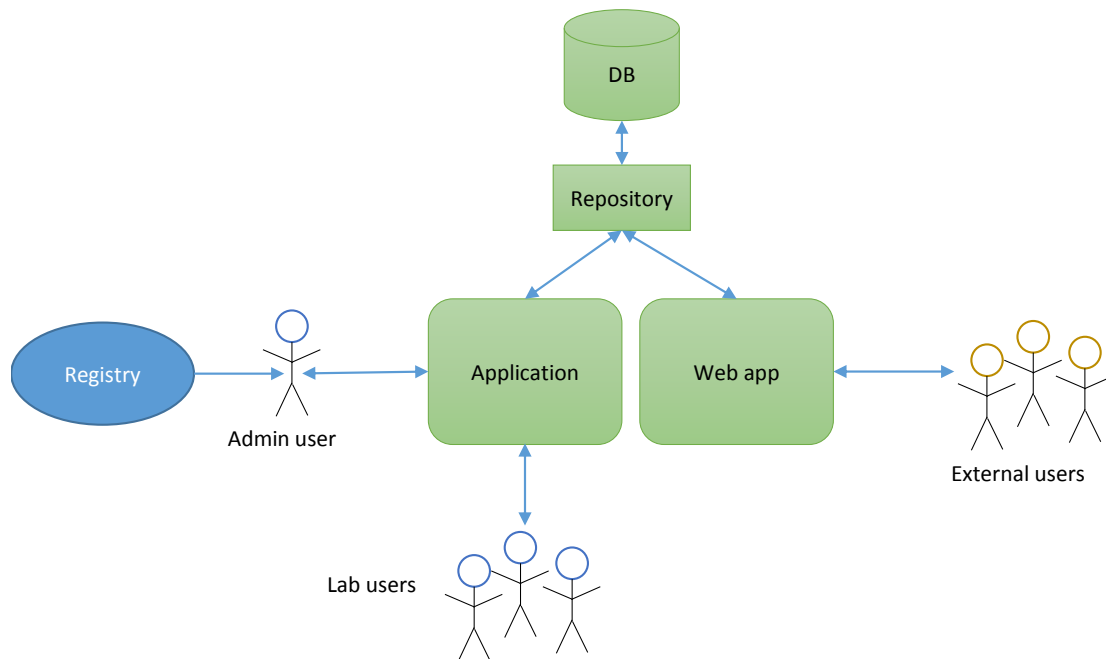


Figure 8: Application model

The Visual Studio solution structure has five separate projects containing five major components of the thesis - application, test, database, setup and web app. The application and web app projects do not depend on each other, they both reference the database project for their database connectivity.

5.5.3 Navigation

The application uses Lim (2011)'s concept of PageViewModels for navigation between the different screens. A PageViewModel is a viewmodel that represents a screen, or page, in the application. She argues that using a viewmodel for keeping track of the application state is a good solution that avoids having navigation code spread across multiple viewmodels. This approach also avoids breaking the MVVM

pattern. The MVVM Light Messenger object is responsible for broadcasting the ChangePage commands.

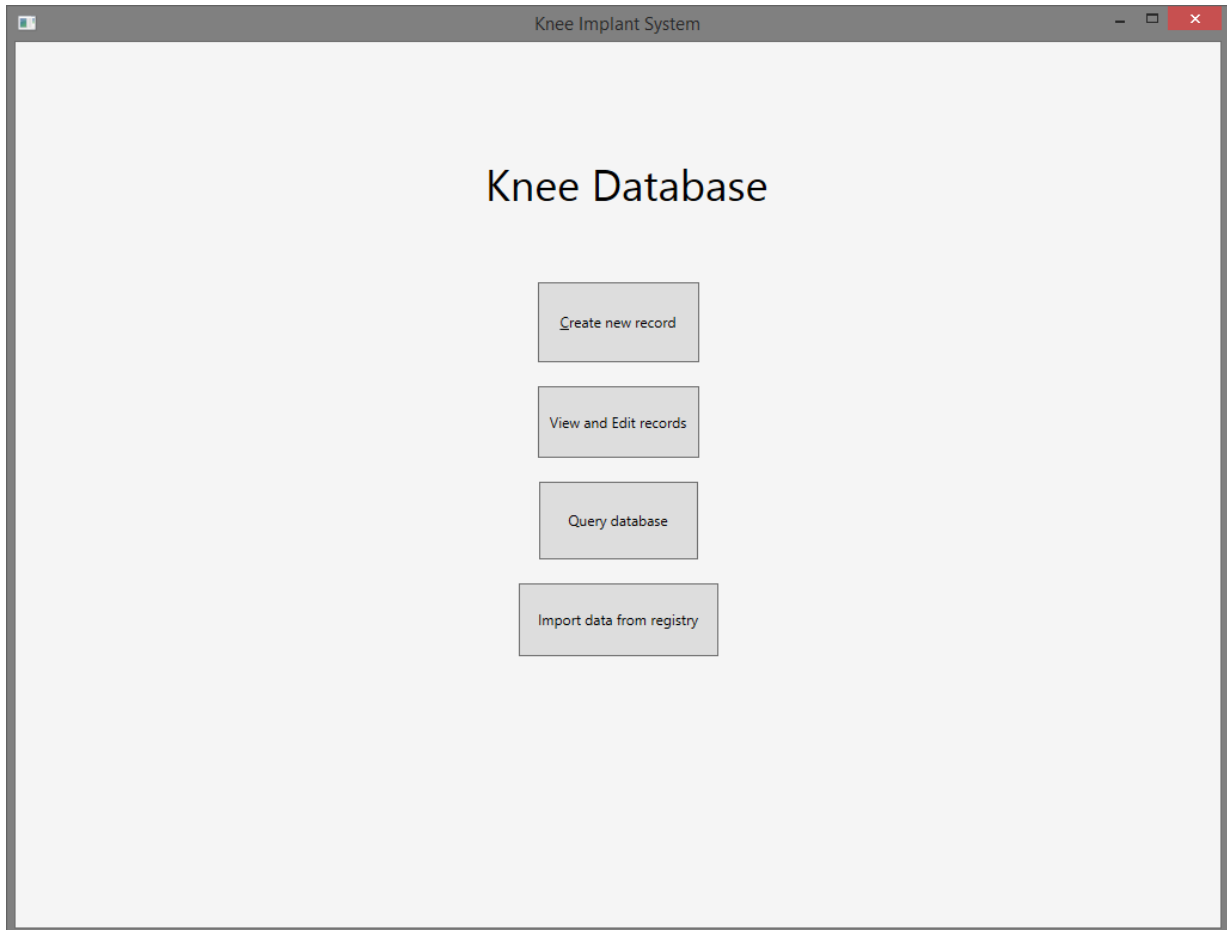


Figure 9: Application main menu

Figure 9 shows the main menu of the application. This is the first screen users meet when they start the program. The different choices here are related to the high level functions they lead to, creating, viewing, querying and importing records.

5.5.4 Record entry

Figure 10 shows the screen where users will create new records when the implants physically arrive at the lab. The information required for entry is entered from

The screenshot shows a software window titled "Knee Implant System". At the top left is a "Main Menu" button, and at the top center is a "Save record" button. The interface is divided into several sections:

- Record:** Contains text boxes for "Record number" (value: 19), "Study", "Record date" (calendar icon, value: 17.03.2014), "Contact surgeon" (dropdown menu with "-- Select surgeon"), "Hospital", and "Created by" (value: Eirik). To the right is a "Picture of components" button.
- Patient information:** Contains text boxes for "Date of birth" (calendar icon, value: 15), "Sex", "Weight", "Height", "Register record no.", "Revision no.", and "Side". There is also a large "Additional info" text area.
- Samples & analyses:** Contains radio button groups for "Radiographs", "Tibial baseplate", "Tibial insert", "Femur", "Patella", and "Synovial fluid". Each group has "Yes" and "No" options. To the right are input boxes for "No. of blood samples", "No. of tissue samples", and "No. of cement samples".
- Implant Removed:** A text box with a calendar icon and the value "15".

Figure 10: Create new record screen

the paper based form filled out by the operating surgeon and shipped with the implants. The screen consists of common user interface controls for data entry such as text boxes, combo boxes, date pickers and radio buttons. Use of well known standard controls should ensure that novice users should still be familiar with the way the data entry works. In the Samples and analyses group, the user is presented with a series of yes/no buttons which must be answered. The hip implant database uses check boxes for this instead, but this posed a problem where there was uncertainty in the case when boxes were unticked. This could have two different meanings - the user knowingly set the value to "no" by not ticking the box, or the user did not answer it at all. The forced yes/no choices in this screen circumvents this problem. This was a feature request from the admin user.

As seen in the screenshot, some of the fields are coloured red to indicate that they require the user to answer them. Unless all required fields pass validation, the save record button is disabled and the record can not be saved. This is done to prevent records that do not contain the minimum required information. Some of the fields display tooltip error messages when hovering over them, such as when inputting letters into fields that require numeric values. In the code listing below, the Weight field is checked to ensure it is not zero or an invalid number, this includes being a character value. Note that most of the code listings in this thesis will be in the C# language, however, any reader with a basic proficiency in a C style language should be able to understand them.

```
if (columnName == "Weight") {
    if (Weight.Equals("")) {
    }
    else {
        try {
            var number = Int16.Parse(Weight);
            if (number == 0) {
                result = "Weight cannot be zero";
            }
        }
        catch (FormatException) {
            result = "Not a valid number";
        }
    }
}
```

Listing 1: Input validation example

5.5.5 Record viewing

As an alternative to navigating by inputting record number or navigating from one record to the next, the users can view a list of all available records and quickly view some key properties such as the status of the record. This screen is shown in Figure 11. Double clicking a record opens the detailed record view. The following code listing exemplifies the page based navigation approach previously discussed.

```
Messenger.Default.Send(new ChangePageMessage { page = "ViewRecord" });
Messenger.Default.Send(new RecordMessage { Record = SelectedRecord });
```

Listing 2: ViewModel communication

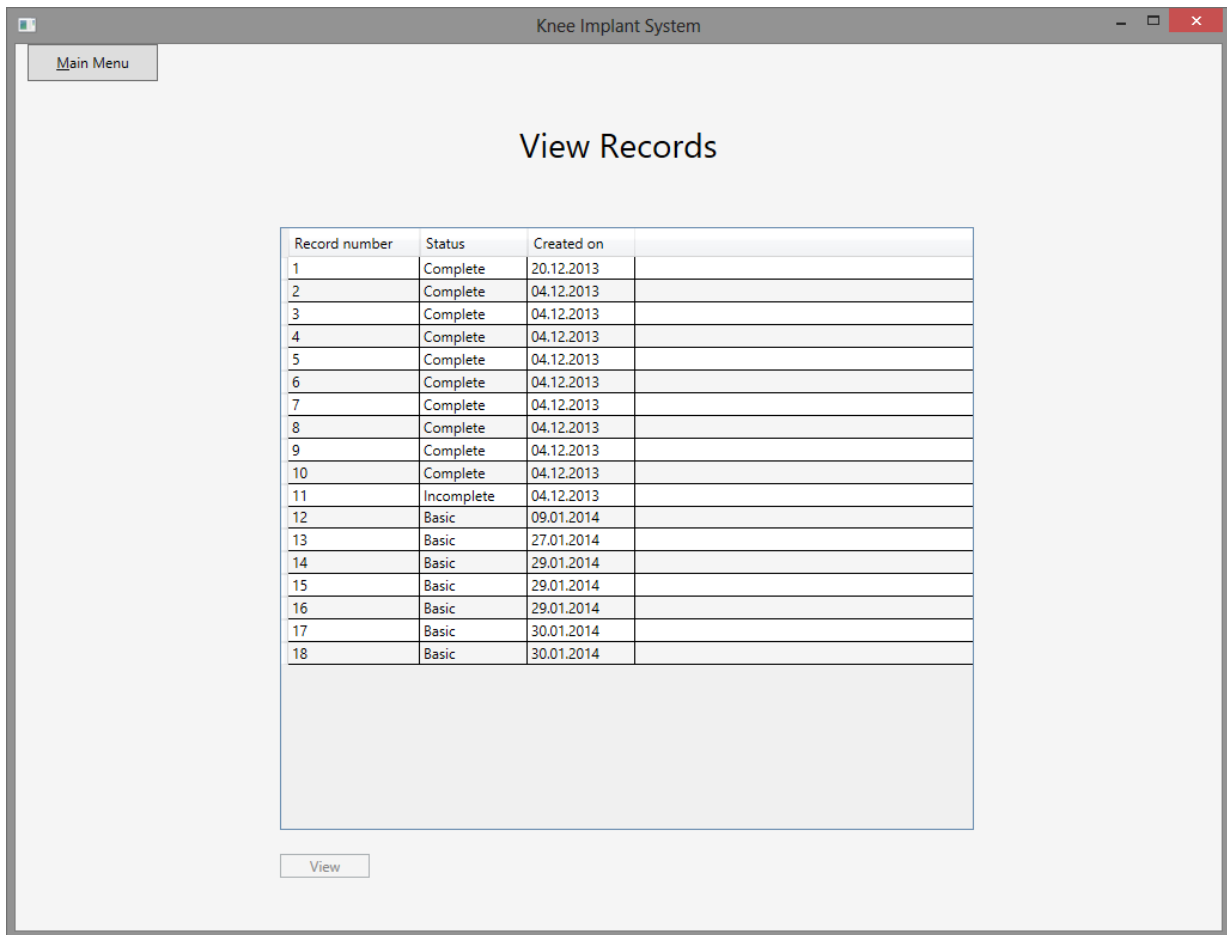


Figure 11: View list of all records screen

A `ChangePageMessage` is broadcasted to all subscribers with the page being `ViewRecord`. Also, a `RecordMessage` is sent to listening viewmodels to indicate which record that is to be viewed.

The detailed record view shown in Figure 12 provides the user with an integrated view of both the information entered by the lab users and the information retrieved from the registry, if available. The most important part of the registry information is in the bottom group named `Revisions`. In this view users are shown the details of the implant components used in the revision that they received the implant from. They can also navigate to earlier or later surgeries when applicable. This is a new feature compared to the hip database which allows users to compare component changes from previous revisions and see which parts were changed

Knee Implant System

Record

Record number:
 Study:
 Record date:

Contact surgeon:
 Hospital:
 Created by:

Patient information

Date of birth:
 Weight:
 Register record no:

Sex:
 Height:
 Additional info:

Revision no.:
 Side:

Samples & analyses

Radiographs
 Synovial fluid
 Tibial insert
 Tibial baseplate
 Femur
 Patella
 No. of blood samples:

No. of tissue samples:

No. of cement samples:

Revisions

Date implant removed:
 Date implant inserted:
 Removal reason:

	Product name	Manufacturer	Cat. number	Size	Serial	Lot	Material	Fixation type
Femur	48							
Femur cement	Palacos R + G	Heraeus Kulzer						1
Tibial baseplate	48							
Tibial insert	48							
Tibial fixation	Palacos R + G	Heraeus Kulzer						1
Patella								
Patella fixation								

Figure 12: View single record screen

and why. Some of the information is currently in the numeric code as they come in from the registry, while other parts are translated to more meaningful string representations. The missing parts were not implemented due to lack of available lookup tables for those components.

All of the fields in the detailed view are disabled by default as a safety feature to prevent accidental editing of the record. Users need to toggle the edit switch to be able to edit the contents of the record.

This screen is also where blood and tissue sample values are accessed from, as shown in Figure 13. These contain a limited amount of functionality compared to the navigation pages and are instead implemented as modal dialog windows as shown in Listing 3. Opening dialogs in a strict MVVM fashion involves imple-

menting a great amount of boilerplate code. For this reason, the opening of dialog boxes is not handled in line with the MVVM pattern, but instead the viewmodel handles creating and opening the view itself. Naturally this breaks with the idea that viewmodels should not be aware of any view components, but in this case purism was sacrificed for the sake of simplicity.

```
/// <summary>
/// Opens a dialog with the available blood samples.
/// </summary>
public void OpenBloodSamples()
{
    var bloodSampleWindow = new BloodSampleWindow
    {
        WindowStartupLocation = WindowStartupLocation.CenterScreen
    };
    Messenger.Default.Send(new RecordMessage { Record = CurrentRecord });
    bloodSampleWindow.ShowDialog();
}
```

Listing 3: Direct way of opening dialog window

5.5.6 Query builder and exporting

The application provides a way to perform simple queries against the database. This interface can be seen in Figure 14.

So far only attributes can be used to filter results - *status*, *surgeon* and *removal reason*. It was a deliberate decision to make the querying system simple, the admin user wanted the application and the database to be more of a data repository than a tool for analysis. Instead, the idea is that once per month a complete export of all the database information would be exported in the form of an SPSS file. This file would be available for the users to perform statistical analyses on. In this way, users would not export their own snapshots of record data and keep isolated local copies. It also decreases the complexity of the application and it would be very time consuming to try to implement even simple versions of the kind of the statistical analyses that specialized software like SPSS does.

The code behind the query builder manages to avoid hitting the database for each query parameter and instead builds up the query arguments one by one and then makes a request to the database with a combined query. A query that filters by a *certain record status*, an *operating surgeon* and a *removal reason* would then

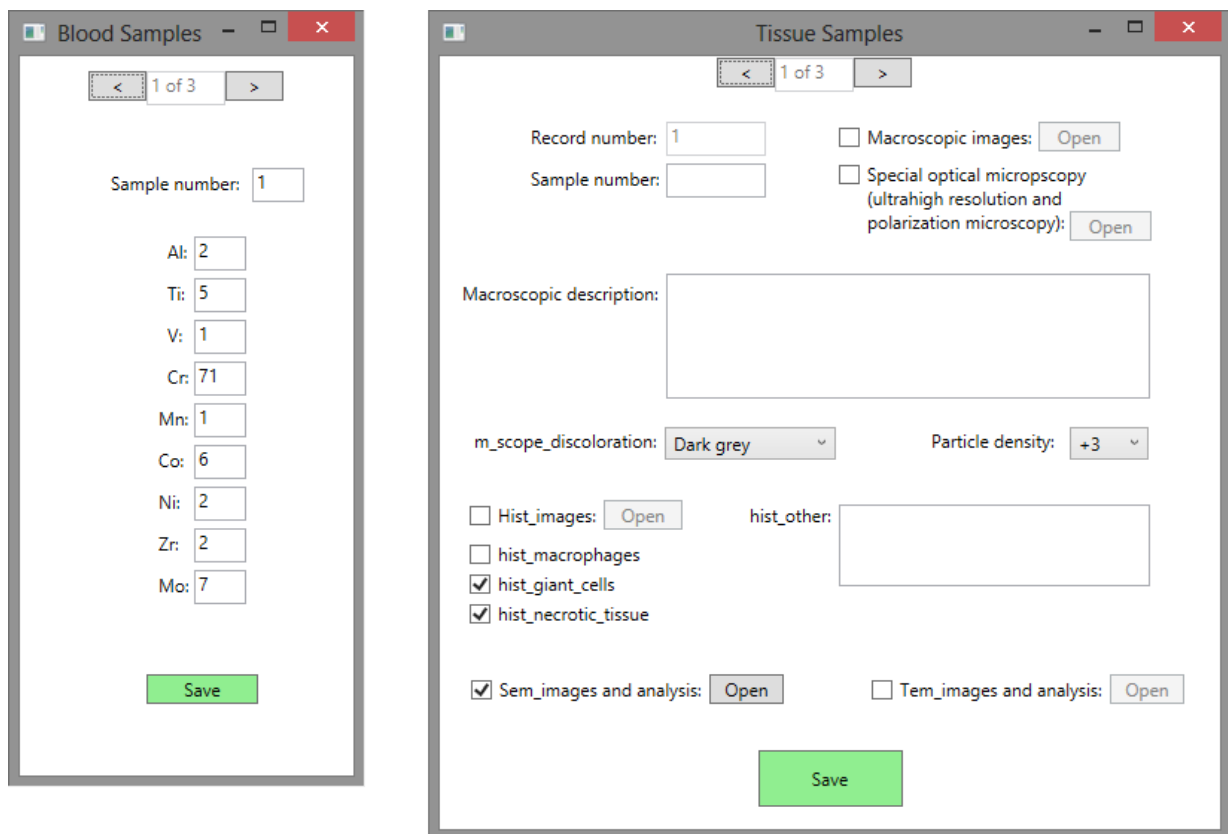


Figure 13: Blood and tissue sample dialogs

produce approximately the same load on the server as a simple query only specifying a surgeon. The code listing below shows how the “filteredClaims” variable is built up incrementally. It does not actually hit the database until the last line where its ToList method is called. This avoids the scenario where the database is queried for each of the arguments and just aggregating the results locally at the end. This modular approach to query arguments also makes it fairly easy to extend the functionality by adding additional query options to the users. The listing below shows how the query method takes in the criteria as parameters and if they are not blank, adds them to the query.

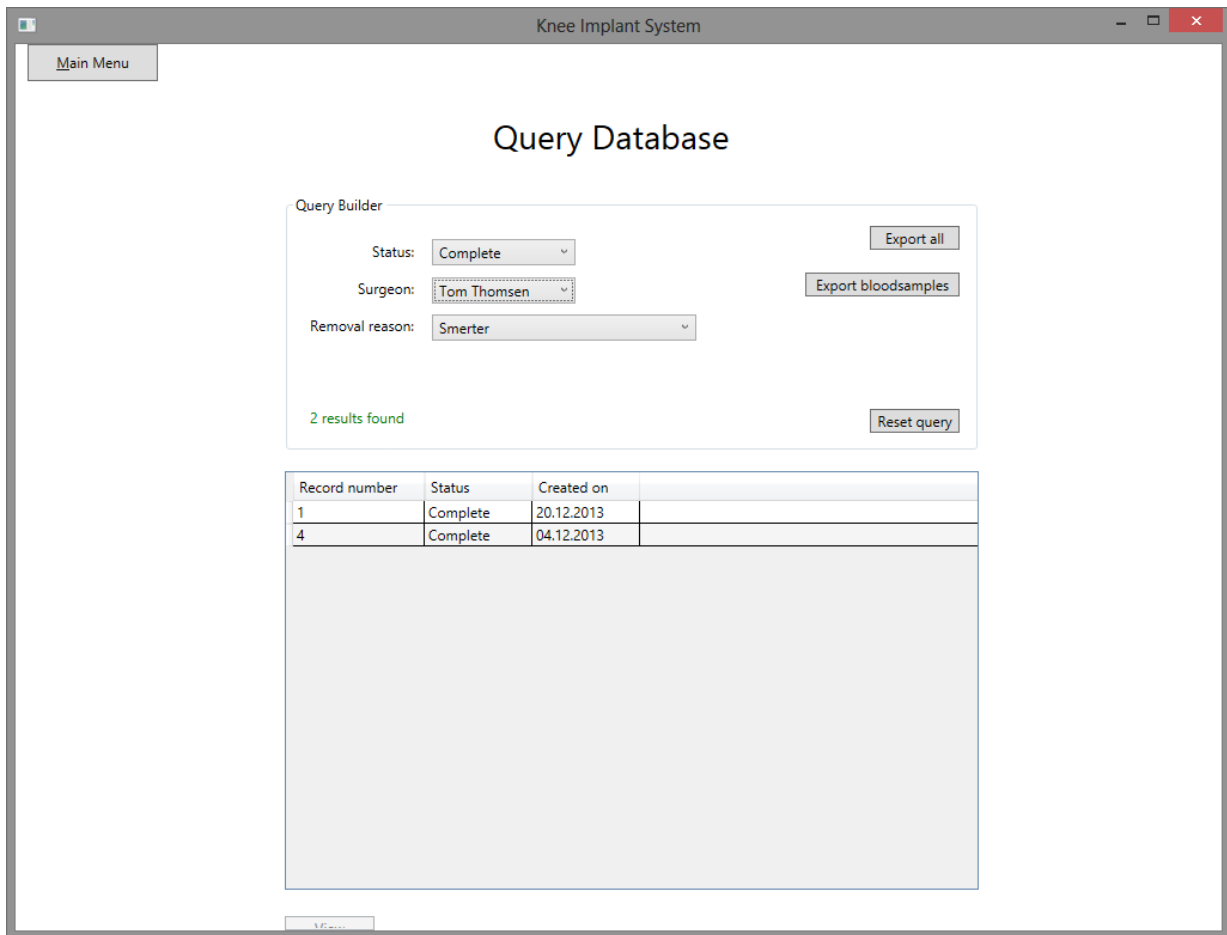


Figure 14: Query records screen

```

public List<Record> GetFilteredRecords(string status, string surgeonName, string
removalReason)
{
    IQueryable<Record> filteredClaims = db.RecordSet;

    if (!string.IsNullOrEmpty(status))
    {
        filteredClaims = filteredClaims.Where(rec => rec.Status == status);
    }

    if (!string.IsNullOrEmpty(surgeonName))
    {
        filteredClaims = filteredClaims.Where(rec => rec.Surgeon.Name ==
surgeonName);
    }

    ...
    ...
    ...

    return filteredClaims.ToList();
}

```

Listing 4: Building the query criteria

5.5.7 Importing registry data

Importing registry data in an efficient manner is crucial to the application. The controls for importing and matching records are shown in Figure 15. The source of the registry data is an SPSS file containing 862 variables and values for each record in the file. All of the values in the SPSS file are imported and stored in the database. They are then searched to see if they correspond to a record already in the database. If it matches a record with no associated registry information, a new association is made. If the association already exists, the values in the registry entry are updated for that record.

Due to the sheer volume of data contained in the registry files, and the fact that they are created by an external party outside the control of the lab users, these variables can not be hard coded to mappings in the database. Instead, a more flexible solution is needed. This issue arises both when importing and exporting records containing registry information. It was solved by using a reflection based approach where the properties of a Registry object at runtime are iterated over and written to rows and columns in the database when importing, or in the output SPSS file when exporting. This avoids the very inflexible and rigid hard coded approach. This is illustrated for record exports in the listing below.

```
foreach (var rec in DataManager.GetRecords())
{
    SpssCase cs = doc.Cases.New();

    PropertyInfo[] recordProperties = record.GetType().GetProperties();
    WriteProperties(rec, recordProperties, cs);

    var reg = rec.Registry;
    if (reg != null)
    {
        PropertyInfo[] registryProperties = reg.GetType().GetProperties();
        WriteProperties(reg, registryProperties, cs);
    }

    cs.Commit();
}
```

Listing 5: Reflecting and writing on record properties

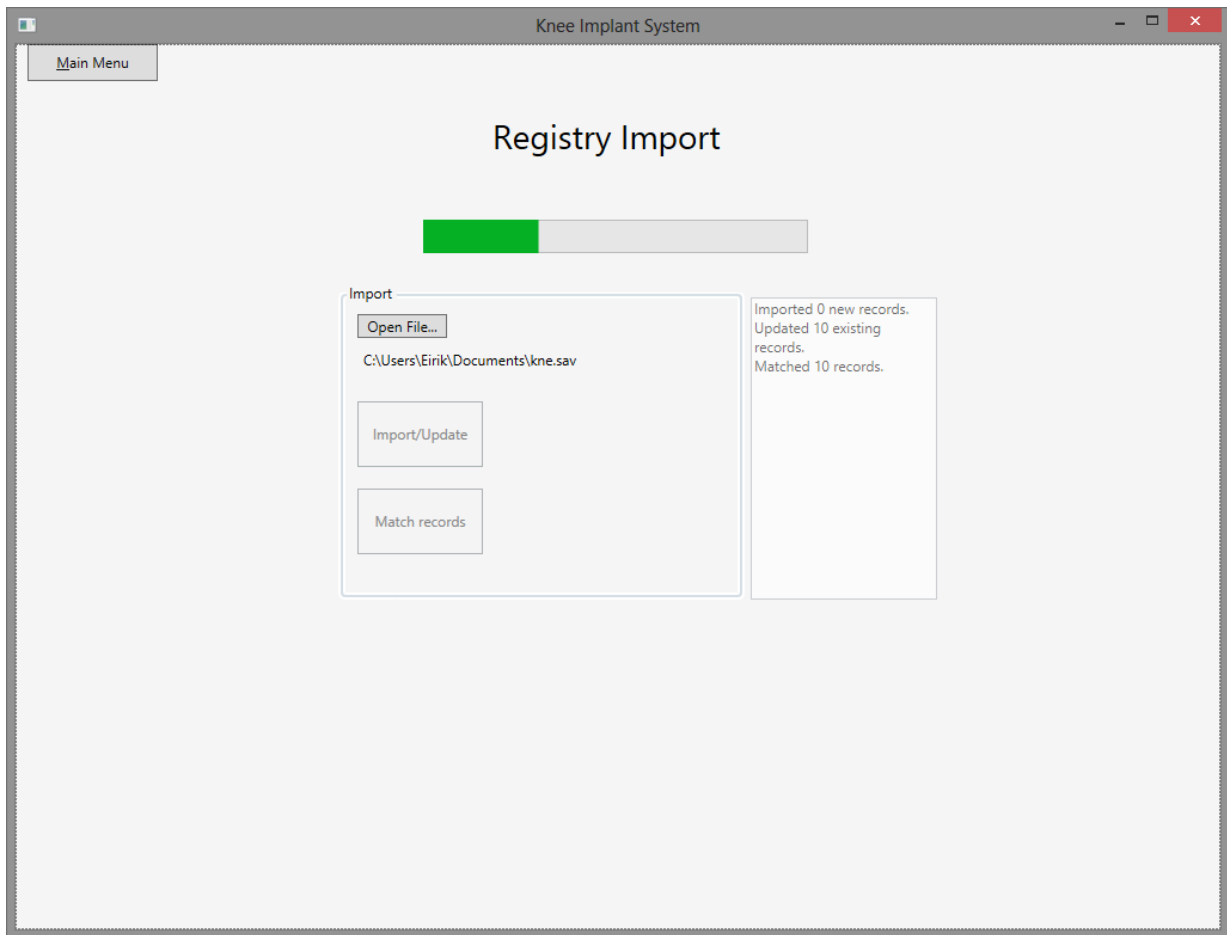


Figure 15: Import registry data screen

5.6 Web application prototype

To demonstrate the feasibility of developing a web application using the same database and model classes as the desktop application discussed above, a basic prototype was developed. This prototype adds a ASP MVC project to the solution, referencing the database project and using it for connectivity, data modeling and input validation. This is not done by directly interacting with the database, but rather through the Repository pattern discussed in Chapter 5.2. It has a limited functionality and user interface as compared to the desktop version, but it is meant as a proof of concept prototype.

As shown in Figure 16, the web application lists all of the records in the

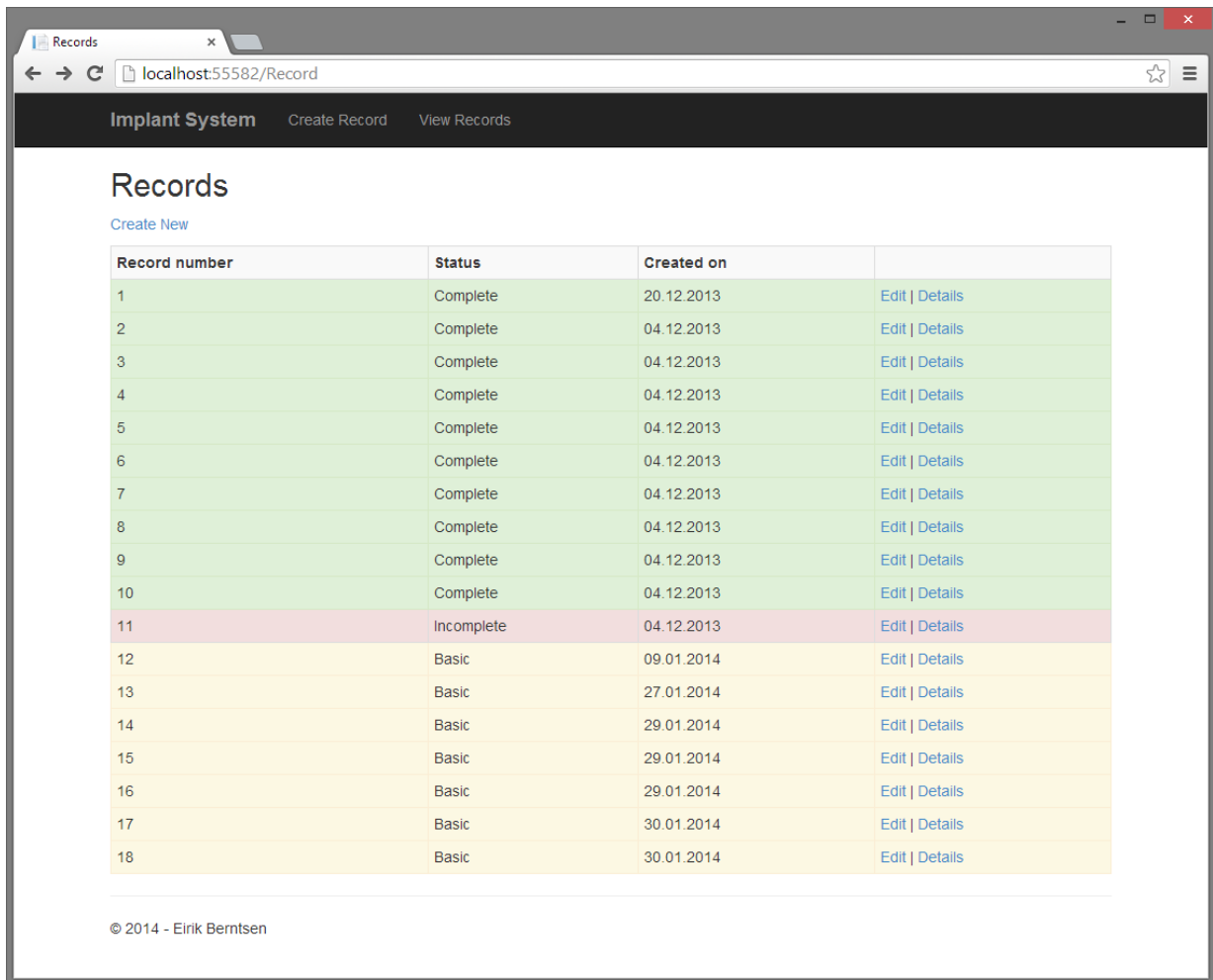


Figure 16: Web application record list

database and colours them according to their current status. The records that lack basic information are coloured red, the records that have the basic information but are not matched with registry data are in yellow. The records that are completely filled out and has been matched with the registry data are coloured green.

Links to either edit or view details about a particular record are shown in the last column. The view also has a link to a basic screen for creating new records.

The detailed record view shown in Figure 17 lists a subset of the properties for a given record, such as the record number and name of the operating surgeon. Many of the fields shown in the desktop application are omitted as this is only

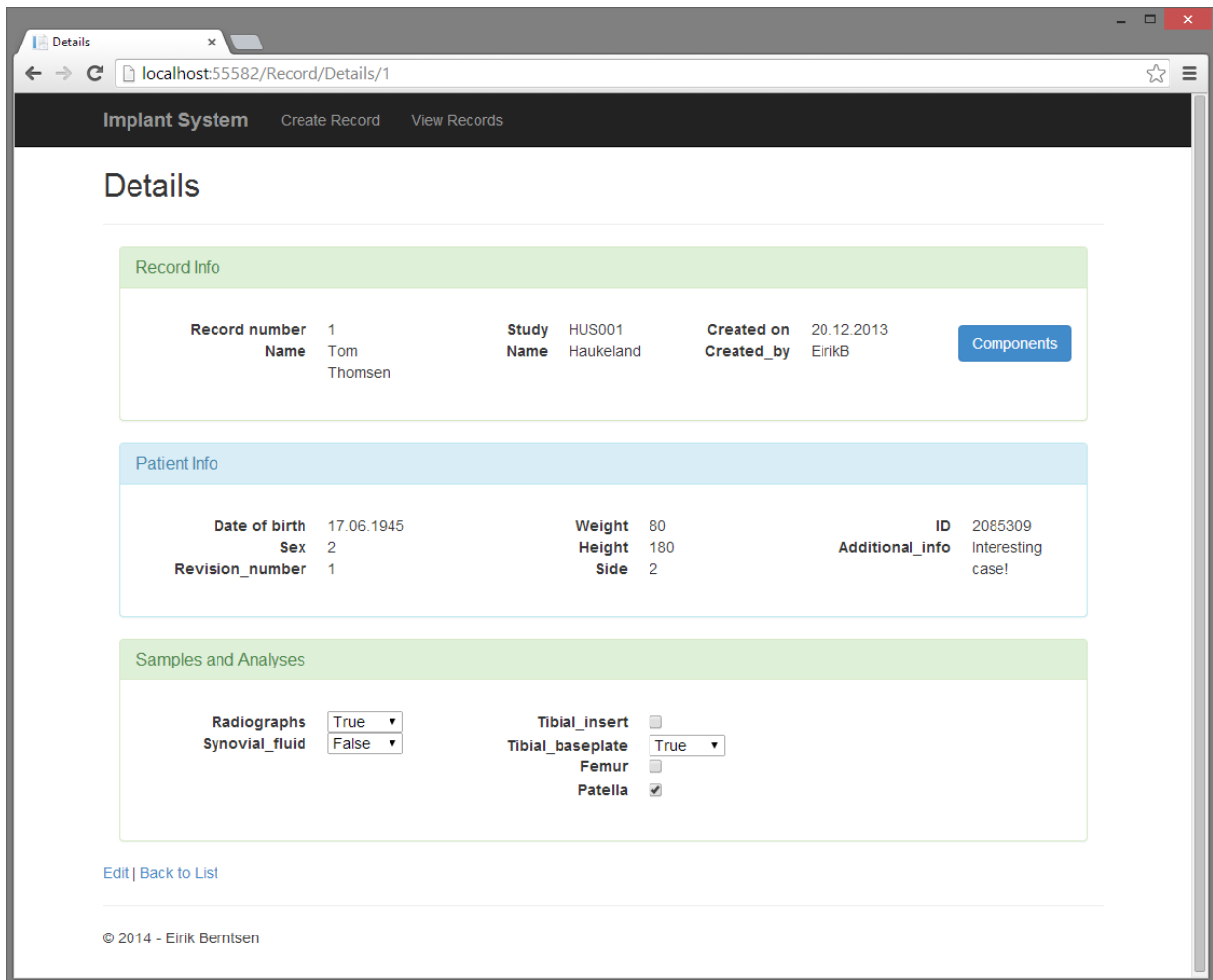


Figure 17: Web application record details

meant as a proof of concept. The administration functionality in the desktop application such as importing and exporting files to SPSS are not covered in this web prototype.

6 Evaluation

Two methods of evaluation were used in this stage of the thesis, user testing and heuristic evaluation. The user testing was carried out with the users at the Haukeland lab as participants, while the heuristic testing used master students at the Department of Information Science and Media Studies as participants.

6.1 User testing

To guide the user testing process, a test plan was created as recommended by Rubin and Chisnell (2008). This plan served to structure the creation of the test tasks, the user testing sessions and evaluation of the test results.

The broad goals of the user testing were to get the opinions of actual real world users on the prototype as compared to the previous solution and to determine if the needs of the different user roles are met.

More specifically, the research questions that the user testing sought to answer were:

- Is the prototype easy to use and to understand?
- Can users carry out common tasks without errors?
- Will use of the system require any special training?
- Where does the prototype need improvements?
- Can the prototype be further developed to fit the users daily needs?

As the total number of users currently working at the lab is only 5, it was crucial to get as many as possible of the potential users involved in the testing session. The main user test was carried out with 4 of the 5 users. The fifth user is the admin user which has such a detailed knowledge of the system due to close cooperation in the design and development phase that the results would be skewed by his participation.

Regardless of the fact that the user group was so limited, research suggests that a sample size of only 4-5 subjects can detect 80% of the usability problems with a system (Virzi, 1992). All the subjects consisted of staff working with hip and knee implant analysis at the Haukeland University Hospital. The test subjects had a gender distribution of 50% female and 50% male. All the participants gave their written consent through the informed consent form shown in Appendix A. The user testing was done in a within-subjects design, which means that all the users were all given the same tasks to perform (Rubin & Chisnell, 2008). This led to a few challenges, as some users are more familiar with some tasks than other users. However, this was outweighed by the desire to keep a sufficiently large sample size which was representative of the working environment.

The tasks were also presented to all the users in the same order. This can lead to an undesired phenomenon called transfer of learning which states that skills picked up in the preceding tasks can affect performance in the later stages of the test session. This can result in tasks occurring later in the test to appear easier to perform than what they really are. Rubin and Chisnell (2008) suggests *counterbalancing* as a way to minimize this effect. To counterbalance the tasks, they are placed in a different randomized order for each test subject. However, they also point out that counterbalancing is not always a solution, as some tasks can have a natural sequence in real life. This is in some ways true of the tasks defined for the user testing sessions. They follow a natural chronological progression - entering a new record, viewing and editing that record and finally making more complex queries against all the records. While this sequence follows the lifecycle of a record, these tasks may not always be carried out in rapid succession like in the user testing. Some users may also only be involved in specific parts of this process. An overview of all the user testing tasks is shown in Table 3.

Task	Description
1	Create and save a new record.
2	Find the date of birth of record 5
3	Find the blood sample values of record 7
4	Find the tissue sample values of record 9
5	Edit the weight of record 4
6	Find the product name of the femur cement component of record 9s second revision.
7	What is the status of record 11?
8	Make a query that finds all records that have Tore Thorsen as the surgeon
9	Make a query that finds all records that have Jan Jensen as the surgeon AND has Smerter as a removal reason

Table 3: User testing task overview

The participants were given the tasks on a sheet of paper, to avoid having to navigate back and forth between the instructions and the application on the computer. The full task descriptions with complete instructions are shown in

Appendix D.

In order to allow screen and web camera recording, a laptop computer was used as the testing device instead of the computers at the Haukeland lab. The lab computers would be a more natural setting, but the ability to record the session was considered to be more important. The laptop was a Macbook Air running Windows 8 with the database application and Morae Recorder recording software (TechSmith, 2014) running. An external mouse was connected to eliminate any potential errors due to touch-pad input.

Successful completion criteria (SCC) were defined for each task so that scoring them would be easier. SCC is a definition of the criteria that has to be met for the task to be considered completed successfully (Rubin & Chisnell, 2008).

The initial system state for the first task is that the system has just launched and is sitting idle at the main menu. Initial system states for subsequent tasks are the states that the preceding task left the system in. Starting each task at the main menu was also considered, but that would interrupt the natural flow of the task sequence. Some tasks have natural transitions between them, such as when one task ends on record number 7 and the next one starts at record number 8. In that case, moving back to the main menu instead of just pressing the Next-button would seem unnatural. Another case is where two following tasks take place on the same screen. Moving away from this screen and then back again would disrupt the natural progression and only serve to confuse the participant.

The two main benchmarks that the testing sessions were looking at were the timing of the tasks and the amount and types of errors done by the users.

The role of the test moderator during the sessions consisted of explaining the procedure of the test session, presenting the informed consent forms and introducing the tasks. The goal was to minimize intervention during tasks, and only to intervene if the users' progress stops or if they have questions.

The data that was collected consisted of various performance measures that were extracted from the screen recordings after the sessions. The users' degree of success on a task was scored using the following scale that is commonly reported in publications:

- Completed with ease
- Completed with some difficulty

- Completed with difficulty
- Completed with severe difficulty and subsequent guidance
- Failed to complete

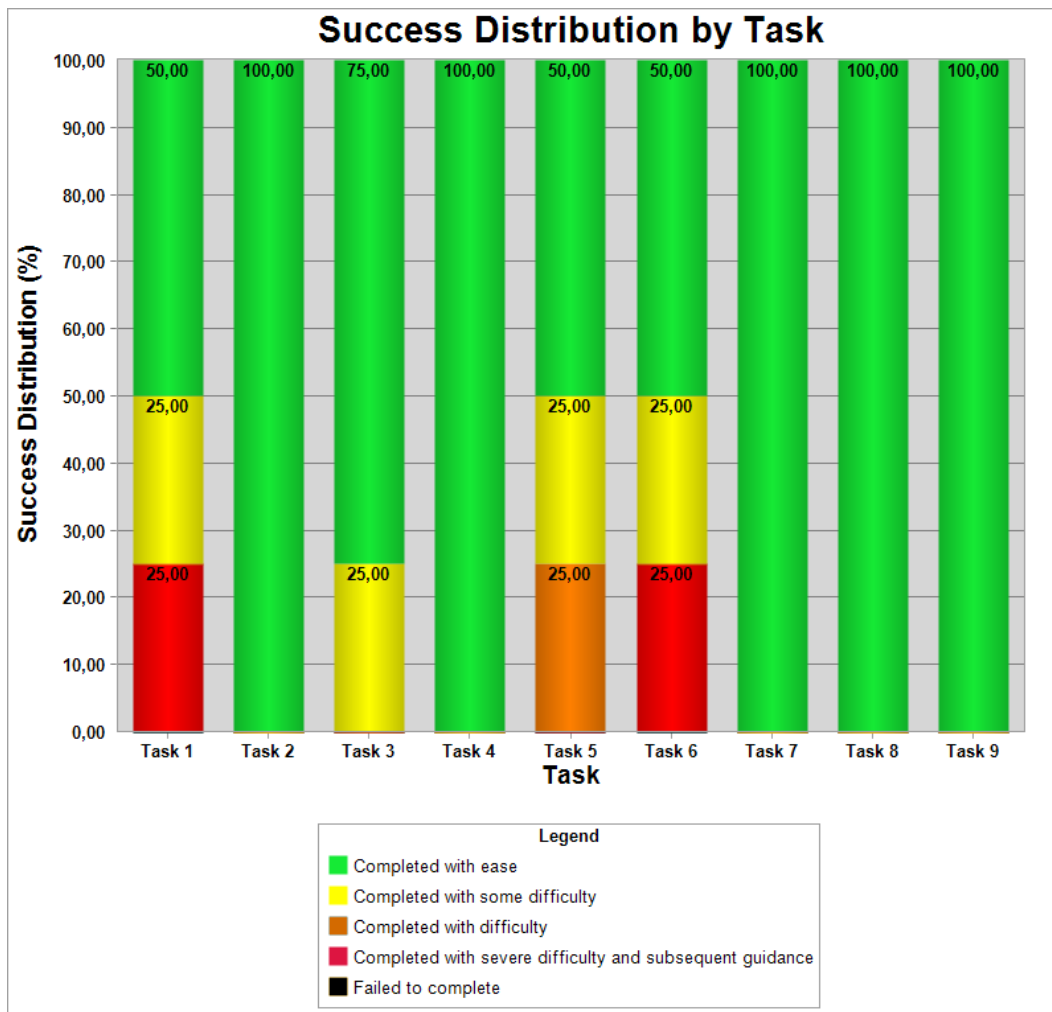


Figure 18: Task success distribution by task

The task success distribution shown in Figure 18 reveals generally positive results, with most tasks being completed with ease and none scored as “failed to complete”. It should be noted that failing to complete a task should be considered an extreme case, as the test moderator intervenes and provides help at the score prior to failure. Task 1, 5 and 6 however, had more than one user experiencing

some degree of difficulty. The difficulties the users were having on task 1 were mostly related to input validation. Some of the users failed to notice the visual cues that show which input fields were required. This result suggests that the prominence of these visual cues should be increased to draw the users attention to them more strongly.

The three main factors affecting task success seemed to be the inherent difficulty of the task, the participants general IT competence and the participants competence on the specific task. Task 1 and 6 were the tasks that were the most complex as defined by the required number of sub-actions needed to complete it. This has been reflected in the success rates.

The general IT competence of the users also proved to be a deciding factor, as some users struggled with user interface controls that follows well established interaction patterns, such as inactive controls being “grayed out” to communicate that they are not currently available for manipulation. To be “grayed out” in this context means that the interface controls that are normally coloured are either temporarily or permanently coloured gray to indicate inactivity.

The third identified determining factor was how familiar the participants were with each specific task. As previously noted, not all the users routinely perform all of the tasks. As expected, some participants struggled with tasks that were not familiar to them from their daily work or from the hip implant Access database.

Another performance measure that was recorded was the average time spent on each task. The results of this benchmark are displayed in Figure 19. These results show a number of interesting trends.

The average time for task 1 was substantially longer than the rest, but this is to be expected as that task involved quite a lot of data entry. This is also a task that some of the users do not perform with the current system and as such are not as familiar with the process.

Tasks 2, 3 and 4 are variations of the same task - retrieving basic information about a record. These three tasks should have the exact same level of complexity. However as Figure 19 shows, the average time spent on these three tasks decreases substantially for each task. A possible explanation for this is that the users are learning from the user interface and are using this newly acquired knowledge to improve their performance in the following tasks.

This pattern is also seen in the last two tasks that are also very similar, but

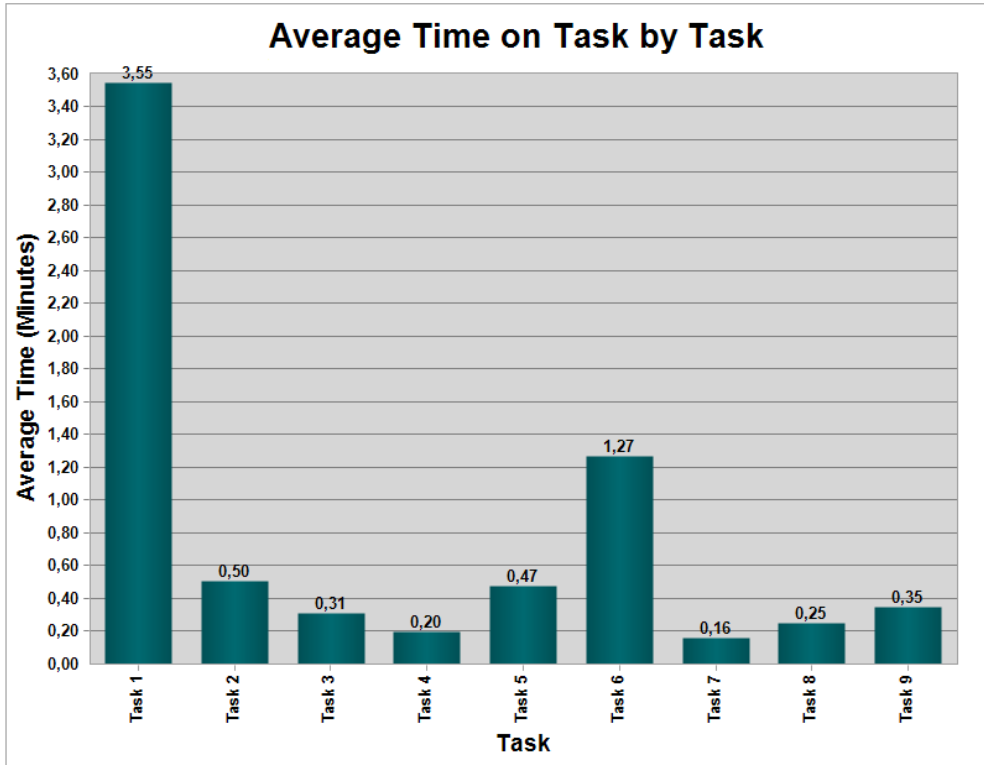


Figure 19: Average time on task by task

the second one has two query criteria while the first task only has one. Despite the second task involving roughly twice the amount of user actions, the average time spent is only increased by 40% instead of 100%.

The results in Figure 19 show the average time on task, but since the number of participants was relatively low and time spent varied considerably between participants, the median and standard deviation for time usage on the tasks is also relevant.

Table 4 reveals that task 1 and task 6 have a significantly higher standard deviation than the other tasks. This corresponds with Figure 13 that shows that these two tasks were the only ones where a user had severe difficulty completing the task and as such spent considerably longer on that task than the other users. The relative deviation is between 31% and 57% for all tasks except the final two, however since

Task	Median	Standard deviation	Relative deviation
1	3,165	1,494	42,12 %
2	0,475	0,208	41,31 %
3	0,31	0,095	31,01 %
4	0,18	0,088	44,60 %
5	0,44	0,155	32,73 %
6	1,2	0,724	57,01 %
7	0,14	0,074	47,05 %
8	0,245	0,030	11,78 %
9	0,345	0,037	10,69 %

Table 4: Median and standard deviations in minutes for task times

the complexity of the tasks and the median performance varied between tasks, this is not generally unexpectedly high. As an example, task 7 had a median performance as low as 0,14 minutes and a high relative deviation of 47,05%, yet the actual deviation was less than a tenth of a minute. This can be attributed to the low number of participants and the relatively low complexity of the task.

After performing all the tasks, the users were presented with a standardized questionnaire - the System Usability Scale (SUS) developed by Brooke (1996).

The SUS questionnaire results in a score ranging from 0 to 100, 0 being the worst score and 100 the best. The average answer scores gathered from the questionnaire are shown in Figure 20 and resulted in a score of 96,25. Figure 21 show the average scores for each individual question. The questionnaire is structured so that a positive response is tied to a high score on the oddly numbered questions and a low score on the evenly numbered. This avoids the situation where a respondent answers every question with a high or low score to try to influence the results in a certain way. While the participants were informed that the survey responses would be anonymous and they were done without their answers being observed, some degree of response bias can not be ruled out. Having a test moderator in the same room sitting opposite the participant causes a much stronger social presence than administering the survey through for instance an online form. In this case, the desire to survey the participants immediately after interacting with the system to get their immediate reactions to the system outweighed this concern. The hope

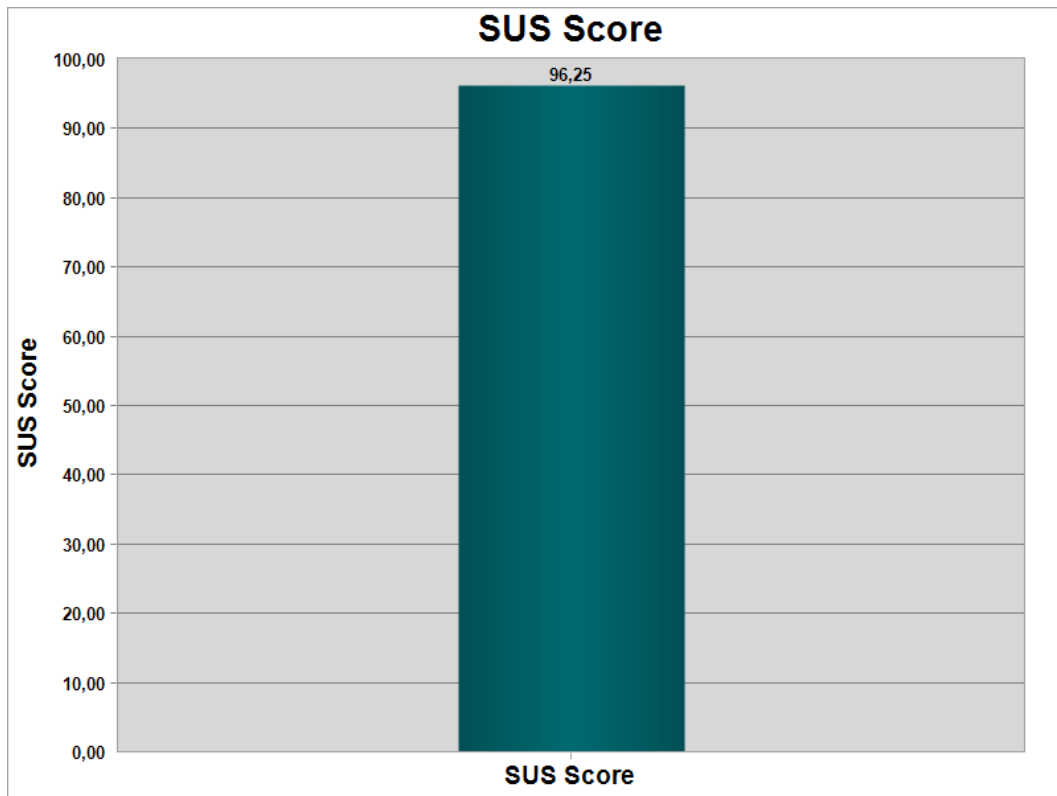


Figure 20: Total SUS score

was that this would capture their most authentic accurate responses when their impressions were still fresh in their minds.

The participants reacted positively to the testing and several of the users stated that it has been an interesting experience.

6.2 Heuristic evaluation

A copy of the detailed heuristics found in Appendix E was distributed to the evaluators in case they were unsure of the specific details of one or more of the heuristics. They were free to consult this form at any time during the evaluation.

Nielsen and Molich (1990) recommend to never rely on a single individual for heuristic evaluation and instead use 3-5 participants and aggregate their evaluations. This approach was followed for the heuristic evaluation of the prototype. Four fellow students were recruited to evaluate the system according to Nielsen's

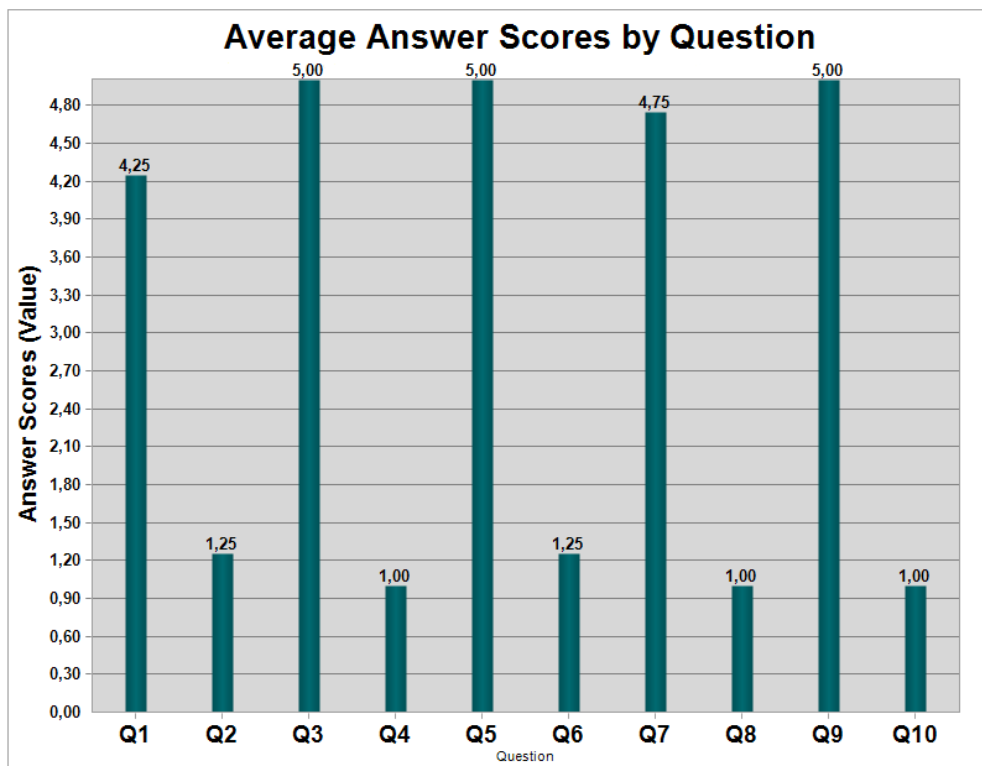


Figure 21: Average SUS by question

ten heuristics. The evaluation session followed the guidelines from Travis (2007) for heuristic evaluation using Morae Manager, the same application used for the user testing. This allowed the task definitions to be easily reused.

Informed consent forms (see Appendix A) were presented to all the evaluators, stating that they agreed that the screen and the computer microphone would be recording during their evaluation. They were informed that these recordings would only be used to categorize the identified usability issues.

The evaluators were asked to perform the same tasks as the user testers were, but they would not be timed. They were explained that this session would not assess their performance in any way, but rather focus on potential usability issues. They were also instructed to ask questions if anything was unclear or confusing, especially concerning medical terms in the application that they could not be expected to be familiar with. Whenever they encountered a usability issue, they were asked to press a key combination on the keyboard and verbally describe the issue.

These events could then be searched and easily organized afterwards. When the events were examined, they were categorized into one of the ten heuristics defined by Nielsen and Molich (1990). If the issue did not fit in any of the predefined heuristics, it would be categorized as miscellaneous, but all of the issues matched a corresponding heuristic.

Nielsen (1995) suggests using a scale to determine the level of severity a single usability issue has. This scale ranges from 0 to 4, 0 being a non-issue and 4 being a catastrophic usability problem.

He defines the severity ranking as a product of three factors. Frequency - how often the issue occurs. Impact - how much of an obstacle the problem is for the user. Persistence - how lasting the problem is. If a user can overcome the problem when he knows about it, the persistence is low. In addition to these factors, market impact is also considered - if the issue will have a negative effect on the popularity of the system. These factors were considered and used to give each discovered usability issue a severity rating. Below is an examination of the problems found.

Issue no.	1
Location	Create new record
Heuristic	Match between system and the real world
Issue	It is unclear if height should be in meters or centimeters.
Recommendation	Height and weight inputs should specify which units are expected.
Severity	2 = Minor usability problem

Table 5: Heuristic issue 1

The first issue is that the height and weight entry fields on the new record screen does not specify which units to input. This is more of a problem with height, where both meters and centimeters are frequently used. The weight can be assumed to be in kilos, but they should still both indicate the appropriate units. Issue 1 is scored as a minor usability problem. It is not purely a cosmetic issue, but on the other hand it is easy for the users to overcome once they see what the expected unit is.

Issue no.	2
Location	Create new record
Heuristic	Recognition rather than recall
Issue	There is nothing indicating the correct date format if a user chooses to type in the date instead of using the date picker widget.
Recommendation	The date input fields should have hints suggesting the proper date format (22.04.2013).
Severity	2 = Minor usability problem

Table 6: Heuristic issue 2

Issue 2 became apparent when one of the evaluators wanted to type in the date manually instead of using the date picker. He was unsure of which date format to use, for instance 23.02.2014 or 23/2/14 or any number of other formatting options. This was also scored as a minor problem because an easy workaround is using the date picker, and like the preceding issue, users only encounter this problem the first time before they learn the appropriate format.

Issue no.	3
Location	Create new record
Heuristic	Consistency and standards
Issue	The save button in the entry form is located in the center top position.
Recommendation	The save button should be in bottom right as it is more in line with other software and follows a natural top-to-bottom flow.
Severity	3 = Major usability problem

Table 7: Heuristic issue 3

Issue 3 was reported by two of the four evaluators who found the location of the save button to be unnatural and they expected it was located in the lower right corner, especially after having worked through the input fields in a left-to-right

and top-to-bottom order. This is scored as a major usability problem as it was deemed as having a higher negative impact than the other issues.

Issue no.	4
Location	Blood sample dialog
Heuristic	User control and freedom
Issue	The save button on the blood sample screen suggests that the user has to make a change instead of just going back to the record.
Recommendation	Either provide a back button or disable save button until changes are made to a value.
Severity	2 = Minor usability problem

Table 8: Heuristic issue 4

This minor issue was reported by one of the evaluators. He felt that having a “Save” button as the only button on the blood sample screen indicated that he had already made a change or that he had to press it to go back to the previous screen. This is not the case, he could have closed the blood sample dialog window. He could also have pressed the save button and the window would close. If no changes were made, the save command would not make any changes to the blood sample values. However, this uncertainty can be easily remedied by the recommendations in the above table.

Issue no.	5
Location	View record
Heuristic	User control and freedom
Issue	When saving a change to a record, it is not expected that the application returns to the main menu.
Recommendation	It should instead either return to the record list or stay at the record detail screen.
Severity	3 = Major usability problem

Table 9: Heuristic issue 5

This issue was reported by all four evaluators and all of them were surprised

to find themselves back at the application main menu after saving a record. There was some disagreement about whether the users should instead be navigated back to the record list view or stay at the record details screen and be given some sort of feedback indicating that the record was successfully saved.

Issue no.	6
Location	View record
Heuristic	Recognition rather than recall
Issue	Record status is not shown directly on the detailed view screen, only in the record list. Requires users to remember the status of the record.
Recommendation	Show record status in the detailed record view.
Severity	2 = Minor usability problem

Table 10: Heuristic issue 6

All but one of the evaluators expected to find the record status on the detailed view screen, but at the moment it is only shown explicitly in the record list view. This is scored as a minor issue because it has a low persistence - experienced users will quickly see that key values are empty in the record. However, providing some sort of visual indication about the records status in the detailed view should be considered.

Issue no.	7
Location	Query database
Heuristic	Consistency and standards
Issue	Dropdown options in query builder are not alphabetically ordered
Recommendation	Order the options alphabetically unless there is another more natural order to the items in question.
Severity	1 = Cosmetic problem only

Table 11: Heuristic issue 7

Issue 7 was categorized as the issue with the lowest severity. It was reported by only one of the evaluators who also commented that he felt it was a very minor

issue. This could be a more important issue if the dropdown menus had a very large number of options in them. In addition to being grouped alphabetically, some menus could be organized by other properties as well, such as grouping surgeons by their associated hospitals. Alternatively, the options could be grouped by how frequently they are used.

Issue no.	8
Location	View record
Heuristic	Recognition rather than recall
Issue	Previous and next buttons on detail screen does not explicitly state that they move between records, not between different information on the same record.
Recommendation	Rename “Next” and “Previous” buttons to “Next Record” and “Previous Record”.
Severity	2 = Minor usability problem

Table 12: Heuristic issue 8

Several of the evaluators said that they expected the “Next” and “Previous” buttons on the detailed record screen would take them between different views of the same record. Because of this they never tried to use them for navigation and instead went back to the record list when navigating between records. Like most of the previous issues, this is not a very persistent problem once the user discovers the way the buttons work.

Issue no.	9
Location	View record
Heuristic	Visibility of system status
Issue	Saving a record does not give any indication of whether or not the record was actually saved.
Recommendation	Provide feedback in the form of a notification or alert.
Severity	3 = Major usability problem

Table 13: Heuristic issue 9

This issue is not exclusive to this particular feature, but is a problem in several

other functions. The application should implement some sort of unified method of providing feedback to the user when actions are performed, such as saving records or other data. This should be done in a non intrusive way. One option would be to display a non modal component that would state the result of some action, “Record saved” for instance. After a period of time it could fade away from the view.

None of the issues reported by the evaluators were scored as a crippling issue, and none of them were considered a non-issue.

After each evaluator had finished all the tasks, they were also given a questionnaire asking to assign a value between 1 and 10 to indicate how well they thought the system performed in each heuristic. The participants were instructed to leave items unanswered if they felt that a particular heuristic was not relevant for this kind of system. The scores from this questionnaire are presented in Figure 22.

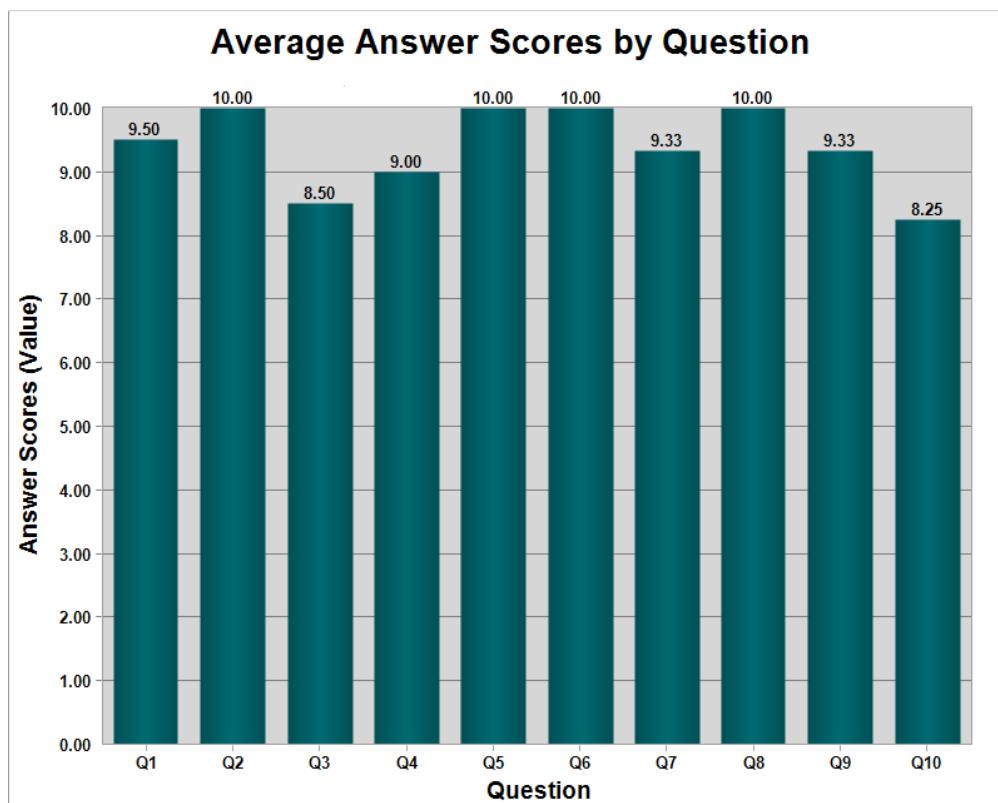


Figure 22: Heuristic evaluation questionnaire scores

Each heuristic was scored by at least 3 out of 4 evaluators which implies that

all the evaluators found the heuristics relevant for the system evaluation and gave the system a thorough assessment.

Perfect scores were reported for heuristics 2 - Match between system and the real world, 5 - Error prevention, 6 - Recognition rather than recall and 8 - Aesthetic and minimalist design. This does not correspond well with the discovered issues, as 4 of the 9 issues discovered were categorized as belonging to one of these top scored heuristics. The explanation for this might lie in the fact that the evaluators themselves did not categorize the issues and might have placed them in a different category. It would be ideal if they categorized them before they were grouped in the most reported category. This highlights one of the shortcomings of heuristic evaluation, its subjective nature means that the results of the evaluation will depend greatly on the evaluators.

Heuristics 1 - Visibility of system status and heuristic 9 - Help users recognize, diagnose and recover from errors, had the second highest scores. This is more in line with the discovered issues, as only one of them were categorized as heuristic 1 or 9.

4 - Consistency and standards, and 7 - Flexibility and efficiency of use also scored relatively high. Two of the issues were categorized as heuristic 4, while none as heuristic 7. Heuristic 4 also shows a slightly lower score than heuristic 7.

Heuristics 3 - User control and freedom, and 10 - Help and documentation were scored significantly lower than the other heuristics. Heuristic 3 represented two of the usability issues and also one of only three scored as a major usability problem.

Heuristic 10 received a score of 8,25 making it the the lowest scoring of all the heuristics. This was interpreted in different ways by the evaluators. Some of them scored it a perfect 10 because they felt there was no need for any help or documentation. Other however gave relatively low scores because of lacking documentation, even though these evaluators also stated that they did not feel the need for it.

In conclusion, the results of the heuristic evaluation consisted of highly positive and constructive feedback for improving the system.

7 Discussion

This chapter includes a discussion of the various parts of the thesis work, starting by discussing the methodologies used. It then looks at the design patterns applied and the prototype itself. The chapter concludes with reflecting on the two different forms of evaluation that was used to assess the prototype.

7.1 Methodology

The term methodology has two different meanings in the context of this thesis. One is the methodology used to guide the design and development work. The other is the broader scoped methodology used to guide the thesis as a design science research project. This section will discuss each of them in turn.

7.1.1 System development methodology

The main choice when deciding on a suitable development methodology in a student project, and indeed in any development project, is choosing between a traditional approach versus an agile approach. Traditional methodologies provide clear structures with predefined deliverables at certain stages in the project (Royce, 1970). They are often seen as very process-oriented methods (McCauley, 2001). Proponents of agile methods claim that this rigidity is more of a hindrance than a help in developing good software. Agile methods are typically more flexible than the traditional methods, in the words of the Agile manifesto - “responding to change over following a plan” (Beck et al., 2001).

This thesis chose to use elements of agile methods and from the Scrum methodology in particular. This was believed to be more suitable for a project of this nature with requirements that were not set in stone from an early point. It also allowed experimentation with ways to adopt typically team-based methods to a single developer setting. The most important of these elements was the user story. Expressing system requirements in terms of user stories rather than detailed documents of functional and non-functional requirements allowed the developer and users to communicate on a commonly understood platform. This was made possible by the fact that user stories are free from any detailed technical structure.

A Scrum-like methodology was challenging to follow strictly in this thesis de-

velopment process. Currently established agile methods might actually not be agile enough for such a non linear process that a student development project often is. Timeboxed sprints with 1-man teams are extra vulnerable to external events. These might include waiting for technical resources such as server space, availability of stakeholders or any personal issues that might occur. Having an on site customer is almost never a possibility for a student project. Many of these issues are also challenges faced by industry development teams, however they tend to have a more structured work week and also a degree of redundancy of resources because of knowledge and skills being shared between members of the team. Single student projects often mix and match practices from agile methodologies and reject the ones that are clearly not suitable for a single developer to follow. Specific methodologies tailored for solo developers, especially in a student setting, are few and not in widespread use. This could indicate a gap in the available development methodologies, or it could be that such a specific methodology is simply not needed by student developers. This issue could be interesting to explore in further detail, but is outside the scope of this thesis.

Despite experiencing some challenges regarding the methodological approach, some of the Scrum elements did indeed serve to guide the development process. In particular, the user stories provided a sense of direction and much needed structure in the implementation phase. A single user story could be used as a commonly understood unit of work when communicating with the users because it is free from any technical details about its implementation.

One might also make the argument that a variation of agile methods was the correct choice for this project, even though it was not a perfect fit. Agile methods as a whole are seen as more nimble and more lightweight than traditional development methods (Abrahamsson & Warsta, 2003) such as the waterfall model described by Royce (1970), and the rigidity of those methodologies would likely be a less flexible choice.

7.1.2 Research methodology

Throughout its different phases, this thesis has striven to follow the design science research guidelines outlined by Hevner et al. (2004) and discussed in Chapter 4.1 in this thesis.

This thesis has produced an IT artifact in the form of a functional prototype, which the guidelines state is the desired result of design science research. The result is a system that streamlines parts of existing processes such as importing and exporting information and providing end users a presentation of this information. However, as the registry stakeholders were not involved in the project, the system does not automate enough of the information flow to define it as a system that is automating a previously manual process. On the other hand, the prototype system does lay down a foundation that could be built upon to enable these processes. Given an arrangement that could secure the privacy of this data across the organization borders, this could be achieved at some point in the future.

Hevner et al. (2004) mention that overcoming potential technology acceptance issues among the end users, organizational and people-based processes might be needed in addition to the IT artifact. Both the organization and the users at all levels have been very supportive throughout this project. They have been willing to both discuss their interaction with current systems and to try out the prototype. While the issue of technology acceptance has been completely absent from the development process, these sorts of issues will probably be more relevant in an implementation phase. The users knew that the testing sessions was a temporary situation and that they would not have to change any current work practices. Social processes such as training might be initiated if a version of the prototype is implemented, but this thesis has focused on the design and development phase and as such is not concerned with such processes in detail. Having said that, the users did perceive the prototype as a system that they could potentially be using, at least some version of it, at some point and their response was largely positive.

The research contribution of this thesis is mainly in the form of the developed prototype, but it also makes a contribution in adding to the understanding of how a student development project can benefit from incorporating elements from agile methodologies and design patterns. The thesis has tried to uphold a high level of research rigor by following well established guidelines and patterns for the different parts of the research and development process.

The guidelines also emphasize the need to present the results of design science research in different ways for different audiences. The resulting IT artifact from this project is presented in a predominately technical-oriented way in this report. At the other end of the spectrum, the prototype has been presented in a more

management-oriented way to the end users. This distinction will be vital to the future of this system. If the work is to continue with the system, a management-oriented presentation of why it is worth spending resources on the system will have to be presented to key stakeholders. This is also essential when considering that there is interest in sharing this data not only locally, but both nationally and internationally.

7.2 Design patterns

The extensive use of design patterns has helped to structure the code base and create code that any potential future developers will be able to understand, use and improve. The use of the MVVM architectural pattern was helpful in directing code design. After building the first few features, the rest of the application follows a set structure and is more predictable and unified. The pattern has also helped keeping the project understandable and organized even when the complexity level was rising quickly. However, this organization comes at a cost. It has resulted in some additional lines of code as compared to applying a simpler approach. In certain situations, following a very strict interpretation of MVVM can lead to a lot of boilerplate code to perform even simple tasks. The prototype intentionally breaks the MVVM pattern in a few places to avoid this problem. The creator of the pattern also points out that MVVM can be unsuitable for simple user interfaces (Gossman, 2006).

Using the repository pattern to create a modular data manager class was extremely helpful when developing the web application prototype. This allowed the web application to re-use practically all of the data storage and access mechanisms that were originally written for the desktop application.

Although the use of the mentioned design patterns may have increased the size of the code base, this increase is well worth it in terms of increased potential for reuse and easier maintainability, especially for new developers. How understandable a piece of software is from a developers perspective is not solely dependent on the number of code lines it contains, but also to what degree that code is structured in a predictable way.

7.3 Prototype

Using a prototype of an application gives users and management a tangible representation of the problem solution, supporting the evaluation of the solution (Budde, Kautz, Kuhlenkamp, & Züllighoven, 1992). This allows for quick, efficient and precise feedback from the customer. When developing the prototype, it was a goal to implement it while respecting processes that are already in place.

The prototype offers two notable improvements compared to using the networked spreadsheet or Access database to manage the joint implant information. One is the ability to see the history of revisions for a given patient. The other is that users are able to view and edit the records concurrently without experiencing write access restrictions.

One major limitation in the Access database for hip implants is that the details of the components used in a revision will overwrite the details of any previous revisions for that patient. This is a problem since researchers may be interested in what types of implants were previously implanted, which parts were changed out and for what reason. The prototype remedies this problem by allowing users to browse between the history of revisions and see details about individual components and the reason for removal. In this way the users can get an idea of the patient's implant history.

Neither Excel spreadsheets nor file-based Access databases are designed to allow concurrent network access from multiple users. If one user has opened the database, the second user who opens it will get a warning that it is locked for editing by the first user. The first user will then have to close the database before the second user can make changes to the data.

The web application prototype will hopefully serve as an example that will enable future students or developers to easily create web-enabled functions that interact with this database system. As an example, one such function might be a data mining function to help report creation based on the data contained in database.

Despite these improvements, some challenges also remain. As seen in Figure 12, some properties are still being displayed in numeric codes rather than in a meaningful string representation. The database tables needed to translate these codes have been made available after the prototype development work was finalized. In

addition to this issue, there are some management-oriented issues to resolve if the system should be put into daily clinical use and not only in experimental use. At the moment the application interacts with a server managed by the UiB IT department. This is a resource that is tied to the completion of this thesis and is being billed to the Department of Information Science and Media Studies. Plans for deployment, backup and maintenance are also important issues that will have to be addressed, but are not within the scope of this thesis.

7.4 Evaluation methods

The two different evaluation methods used in this thesis served complementary purposes. The user testing examined how the real world users interacted with the system and how well it supported their information flow as compared to the current practices. Including both forms of evaluations generated feedback regarding functionality, usability, user experience and technical issues.

Performing user testing of the prototype served several purposes. Besides generating feedback and performance metrics, it also served as a demonstration for many of users. Prior to the user testing sessions, only some of the users had seen earlier versions of the prototype.

Heuristic evaluation used expert evaluators with more technical insight to discover a number of potential usability issues that were not explicitly reported by the user testing. Due to the evaluators having an understanding of the technological standards, limitations and possibilities for such an application, they were also able to offer suggestions on how to improve these identified issues.

The amount and quality of the feedback and constructive criticism received from both the user testing sessions and the heuristic evaluation have exceeded expectations. Each method provided valuable insights that could be transferred to any potential further development. The results gathered from the evaluation offers concrete suggestions for improvements, and it is clear why both these methods are frequently used by software developers.

8 Conclusion

This final chapter will summarize the entire project. It discusses how the initial research questions have been answered by this thesis and finally outlines options for further development of the prototype.

This thesis has shown how a project that is rooted in the information science field, can make use of design science research to develop useful computer based artifacts that targets problems in an unrelated scientific domain such as joint implant revisions.

The current prototype has the potential to impact the information management at the Biomaterials lab. The application can export records to SPSS format, but additional formats for other types of specialized software could also be added. In this way, the researchers could use the database as a central repository and export the data in an appropriate format for a particular analysis.

Using a full fledged relational database instead of networked spreadsheets, or file based databases such as Access, opens up new possibilities in data sharing, both within the organization and potentially with external parties.

Student development projects differ from industry projects in a number of ways. Scale, team size, available resources in terms of manpower, finances and knowledge are all measures that will be significantly different for a student project. However, these differences are not accounted for by development methodologies and there is a lack of specialized methodologies that are suited for such projects. Many student development projects are completed successfully by not using any sort of concrete methodology, but they are missing out on the structure and quality assurances that such methodologies can offer to any project. Despite of this, agile methods also offer many different practices and guidelines that are useful for single developers.

Adhering to well established design patterns both at the architectural level with the MVVM pattern and at the more specific code level with the Singleton and Repository patterns has hopefully resulted in an understandable and intuitive code base. Other student developers who have explored the code informally have had seemingly few problems in understanding the structure of the project, despite its relative complexity.

The code behind the prototype still breaks these patterns intentionally in a few places to avoid unnecessarily complex sections of boilerplate code to perform

simple operations. The aim was to do this in a consistent way so that any other developer will be able to quickly see in which situations this occurs and understand the logic of the code.

If other master students will utilize the code base from this prototype, this can be seen as a kind of team based approach, even though the “team” is separated in time.

Following an established research paradigm such as design science should ensure that the project is carried out in a consistent and predictable fashion so that the traceability of the research remains high.

The developed application that has potential to be further improved to the point where it can be put in daily use and improve the researchers’ data storage and administration. The prototype demonstrates several improvements compared to the existing solutions for both the knee and hip implants. Strategies are being developed of how it could be included in the hospital information system (HIS).

8.1 Future development

During the evaluation phase, the users saw that the system could include some international data sources and provide additional functionalities that would allow information retrieval and exchange across the borders of the institution.

Another master student is currently exploring the possibility of writing his thesis on this same topic and is considering developing a web application to provide access to some of the information in the database to external parties such as surgeons. The prototype would then hopefully be a good starting point for implementing additional features and improving existing ones. With the web application prototype in this project I have shown the feasibility of implementing a web application that uses the same mechanisms for data storage and retrieval that are already defined in the Database project of this thesis. However, the web application prototype in this thesis is built as a proof of concept and disregards important security issues, both technical and administrative, that will need to be addressed. Such issues have to be addressed within the hospital information systems as the Haukeland University Hospital.

The hip implant data could be migrated to the same system and be stored in a unified manner together with the knee implant data. The current prototype could

be altered to use inheritance concepts to re-use much of both the data storage and processing across both implant types as they share many characteristics. In this way a lot of domain terminology present in the prototype could be used by a new developer to get familiar with the domain.

As a possible future extension to the prototype, a developer could expose an Application Programming Interface (API) to external parties that want to utilize the data generated and integrate it with their own.

Another possibility is to provide access to the system to other interested parties such as medical students for educational purposes or even the general public.

This would require tiered access levels with external user groups only having read-only access to certain parts of the information contained in the database.

Appendix A User testing informed consent form

USER TESTING INFORMED CONSENT FORM

STUDY ADMINISTRATOR: _____

PARTICIPANT: _____

This session is a test of the functionality and user interface of the Knee Implant Database prototype. The goal is to help make the system more useful and user friendly and your participation will help in reaching this goal. This test is intended for potential users of the finished system.

In this testing session you will be asked to perform certain tasks with the system. Remember that this is not a test of your performance, but the usability of the system.

In order to analyze the results of the testing, the session will be recorded, both the computer screen through recording software and the participant through a web camera. At the end of the session, a short questionnaire will be presented.

Your participation in this study is completely voluntary and you can at any point withdraw from the session without stating a reason. The results gathered from this session can be used to improve the system and also be published in a report. The findings will not include your name or any other form of identifying properties.

Statement of informed consent

I have read the description of the study and of my rights as a participant. I voluntarily agree to participate in the study.

I understand that recordings will be made of my session. I grant Eirik Berntsen permission to use these recordings for the purposes mentioned above.

NAME (IN PRINT): _____

SIGNATURE: _____

DATE: _____

Appendix B Interview transcript - User 3

Wednesday September 4th kl 13:00 - Haukeland university hospital, office

E: Eirik Berntsen

U3: User 3

E

Først og fremst, hva er stillingen din her?

U3

Ja, de siste tre årene så har jeg vært forsker, finansiert av noe som heter bergen medisinske forskningsstiftelse - bmfs. Det er et sånt rekrutteringsstipend som gis til de som er ferdig med postdoktorperioden sin. Jeg søkte på en utlysning som gikk på translasjonell forskning, det vil si basalbiologisk forskning på et klinisk materiale. Vi liker å kalle det... at vi ser på... at vi tar materiale fra klinikken til labben. Og så forsker vi på det, og så kan eventuelt resultatene gå forhåpentligvis gå tilbake igjen til klinikken, til pasientenes beste. Så jeg var heldig å få det for tre år siden, det går ut om en måned. De har heldigvis da lyst ut en stilling her på instituttet, som jeg håper jeg får. Så jeg har tittelen forsker ja.

E

I forbindelse med det du bruker systemet til, hvilke arbeidsoppgaver bruker du disse databasene til? Eller databasen?

U3

Nei, det er jo som en kilde til de studiene vi gjør. Så... Av og til så samler vi materiale med tanke på en studie, der vi skal studere den protesen fordi den svikter, sånn og sånn. Det var vel litt sånn det begynte, vi hadde et spesifikt produkt som viste høy havarirate. Men så fant vi ut at det var kanskje lurt å samle alle protesene som havarerer med en viss grunn. Slik at vi senere kan bygge opp et så stort materiale at vi kan drive forskning på det. Så... det vi gjør er jo ofte å definere en problemstilling, så vi sammenligner tese A or tese B. Vi sendte inn en artikkel i forrige uke, men den designen da. Begge var titanproteser, men den ene var festet med beinsement og den andre var usementert - den gror direkte i beinet. Vi sammenlignet de to gruppene da. Da bruker vi alle de dataene som vi har samlet for å sammenligne gruppevis.

E

Ok.

U3

Jeg vet ikke om jeg skal komme inn på hvilke typer data vi har, eller om vi skal vente litt med det?

E

Ja, det kan vi forsåvidt ta etterpå. Men det er helst slik at du henter informasjon ut av databasen da, eller..

U3

Ja

E

Bidrar du og med data tilbake igjen?

U3

Ja, først så fyller vi jo på med data, da prøver vi å fylle alle boksene kan du si, slik at datasettet blir komplett for alle casene. Men senere kan det hende bare at vi bruker halvparten av alle de opplysningene som er der. Det kommer an på hvilken problemstilling vi har, og hva aimen blir i forskningsprosjektet.

E

Ja. Det å fylle inn data i databasen som du sier, er det noe alle gjør eller er det noen som har det som ansvarsområde?

U3

Ja, det er litt oppdelt. Eh... <User 1> er gjerne den som mottar pakken fra sykehuset, der alle komponentene ligger da. Så hun kan du si oppretter en ny record, en ny side i databasen. Og da begynner hun å tildele sånn type record number, som da er løpende, stigende. Og så fyller hun da inn de første delene av databasen, som er basert på opplysninger som legen har fylt ut i et skjema som vi har gitt ham. Tror vi kaller det implant retrieval form.

E

Mm.

U3

Altså et skjema.

E

Så hun sitter da med papirene og mater inn disse dataene?

U3

Ja, ja. Så det går helt manuelt. Hun tar gjerne bilde av alle de komponentene, som også blir lastet inn i databasen, så ser vi på et bilde hva vi har, av materiale. og det er jo da selve protesen som kan bestå av flere deler, det er vevsprøver - som kan bestå av flere rør, og det er gjerne en blodprøve. Det er en typisk sånn pakke.

E

Ehm.. og så vil hun de ta hånd om de delene og lagre de der de skal, med dette løpenummeret. som har noen sånne underkategorier som... vev blir record 1-T for tissue og så et nummer bak. Og blodprøven får en B, mens implantene tror jeg bare har løpenummeret skrevet på posen.

E

Så nå snakker du rent fysisk lagret...

U3

Ja, men de samme kodene finner du igjen i databasen

E

Ok, så du kan komme tilbake med et implantat og..?

U3

Ja. Og så taste inn det nummeret og så komme akkurat til den recorden.

E

Gjøres det med noe.. er det noe strekkode system eller noe sånt, eller er det bare...?

U3

Nei, det har vi ikke innført. Det er helt skrevet... Det er helt håndskrevet. Men vi har absolutt tenkt på strekkoding, som fremtiden. Enten QR-koder eller... Nei hva heter det?

E

Joda.

U3

Det er QR?

E

Ja, disse firkantete kvadratiske du tenker på?

U3

Ja. Vi har fått sånn QR-kode som symboliserer forskningsgruppen, hjemmesiden vår.

E

Ja okay. Ja, jeg ser den.

U3

Midt på døren der, har du leser så kan du laste opp den. Eller du kan bruke vanlig parallele strekkoder. Det tror jeg produsentene av implantatene gjør. Der er det vanlig strekkode. Du kan sikkert kjøpe klistremerker som du fester på posene for eksempel.

E

Hvor ofte vil du si at du bruker databasen?

U3

Eh... Ja det går jo veldig i....

E

Er det daglig eller er det...?

Perioder da...

E

Så det kan variere?

U3

Ja, men det er ihvertfall ukentlig. Ikke daglig. Når du trenger dataene i studien så er det jo daglig, absolutt. Ja, hvor ofte får vi retrievals? Ikke veldig hyppig, men.. Snittet er ikke mer enn to i måneden eller noe sånt. Det har dabbet litt av i det siste. Vi har fullført endel studier og da ber vi ikke om flere implantater i den kategorien.

E

Snakker du da både om hofta og kne?

U3

Ja, samlet. I det siste har det vel vært mest kne vi har samlet. Det blir liksom den neste studien.

E

Det lå jo ganske mange flere hofta-records enn kne. Er det fordi dere har begynt med det tidligere eller får dere mer av det?

U3

Nei, vi får mer av det historisk sett, men vi begynte også bare med det. Og så begynte vi med kne, for et par år siden eller noe sånt. Og vi har faktisk bare samlet en kneprotese hittill ??.. Så det er veldig sånn studieavgrenset da.

E

At det er til et bestemt formål?

U3

Ja. For det er akkurat den hun, <User 2>, hun gjesteforskere som kom nå i høst, skal studere da. Og det er en veldig mye brukt protese internasjonalt, men den har vist litt skuffende resultat utifra forventningene, for forgjengeren var veldig bra, men nå gjorde de noen "forbedringer" som viste seg å være en forverring i praksis. Vi vil nok sikkert samle 100 kneproteser også. Men da det er nesten sånn at vi må kontakte kirurger og sykehus spesifikt og si at nå ønsker vi den og den type proteser.

E

Så det er ikke slik at...

U3

Vi har ikke bedt dem om å samle alt og sende, da hadde vi fått mye mer.

E

Så de kommer ikke uten at dere ber om det?

U3

Nei, vi sendte ut brev til alle de sykehusene som brukte den protesen, den kneprotesen. Noen sa de ikke var interessert å delta. Andre sa at de hadde veldig lite revisjoner. Andre var positive, de da som har levert til oss. Så det er litt sånn variabel oppslutning om denne tankegangen. Så... Jeg kunne gjerne sett at det var litt større entusiasme blant klinikerne. Det er jo litt jobb, det er ekstra jobb for de, men vi mener jo at de får mye igjen for det i form av resultater. Så...

E

Et annet spørsmål, hvor godt synes du dagens system og hoftedatabase støtter det arbeidet du gjør?

U3

Ja....

E

Er det noen åpenbare fordeler med det, med at du...

U3

Ja, det er veldig oversiktlig, du har alle data samlet på en plass. Det var jo jeg som begynte å samle implantater før <Admin user> kom faktisk. Og da hadde jeg det i ulike system, så bilder var lagret på en plass og mesteparten av dataene var lagret i et vanlig excel-ark med veldig mange kolonner. Andre data var lagret i SPSS for statistisk analyse. Så alle de kvantitative dataene var lagret i statistikkprogrammer. Og ja, det må vi jo idag også, ta ut et datasett for statistisk analyse, men det er jo greit å gjøre med Access, at du kan eksportere data til andre systemer.

E

Hvilken format er det som regel skjer i da når dere eksporterer fra Access?

U3

Nei jeg tror kanskje oftest at du tar det via et Excel ark og så importerer det, men du kan og tror jeg eksportere direkte til SPSS, men det er jo rader og kolonner, så du kan jo bare lage en tekstfil eller ascii fil, så kan du jo importere hva som helst. Men såklart, du kan se litt på om det er mulig å gjøre det.. mer direkte og raskt tilgjengelig for statistisk analyse. Det er et program vi bruker veldig mye for å produsere grafer, som du ser på veggen der for eksempel, det er GraphPad. Som og har en statistisk analyse-del som oppfyller de vanligste behovene våre. Som vanlig T-testing og ja, gruppesamling da, for å se om ting er signifikant forskjellig. Ja, så vi har brukt mindre og mindre SPSS i det siste etter at vi tok i bruk Graphpad. Og de har jo sitt filsystem igjen, som er litt... eh.. vanskelig å eksportere og importere ifra. Men det er litt å importere en excelfil inn i graphpad da, men det er vanskeligere å ta dataene ut fra graphpad igjen, men det er sjeldent vi gjør, så det er ikke noen stor sak.

Så... ja. Det var jo et eksempel da. Jeg synes det er veldig kjekt å ha tilgang på bilder for eksempel, det kunne en jo ikke for eksempel i excel så lett ha. En kunne jo ha en link fra Excel, men her kan du åpne det...

E

Så du åpner mappen, sånn at ...

U3

Ja. Så får du veldig kjapt tilgang på det. Vi har røntgenbilder her. Nå var det ingen på det her caset her, men du åpner undermapper da. Det var et dårlig eksempel.

E

Så det varierer hva som er tilgjengelig av bilder, utifra...

U3

Ja. Det jeg legger inn oftest er jo blodprøvedata, der er jeg ansvarlig for de analysene. Da får du opp en sånn liten tabell. Så langt så har jeg måttet taste de inn manuelt, så det kunne jo vært litt enklere da.

E

Hvor kommer disse tallene fra? Hvor taster du de fra?

U3

Ifra en Excel fil.

E

Okay.

U3

Så jeg får en excel fil fra instrumentet, eller en ascii fil som jeg importerer til excel. Og så må jeg gjerne gjøre litt justeringer, trekke fra blanke og litt sånt. Kvalitetssikre dataene. Så må jeg da plukke titankonsentrasjon, ?? krom i denne rekkefølgen her. Den står gjerne ikke i den rekkefølgen i excel arket. Den står gjerne ikke i den rekkefølgen i excel arket, der gjør jeg det manuelt. Men det er jo ikke veldig mange verdier for hvert case, så det er jo ingen stor jobb. Så...

E

Ja, da beveger vi oss litt inn på det jeg hadde tenkt å spørre om nå, om du ser noen områder som har potensiale for forbedring i dagens system? Du nevnte dette med eksportering til andre programmer, og importering av ...

U3

Ja. En kunne jo lage en type mal i et enkelt system som excel, som gjorde at du importe den rett inn her da. Men da må du jo kanskje...? Vet ikke om det er en forbedring, du må jo fylle ut den malen også så må du importere det i den malen. Så det handler vel litt om det de kaller datamining, plukke de rette tallene fra en annen kilde, det krever vel at du har det i et riktig format i utgangspunktet.

E

Ja... At de er konsistente, ikke nødvendigvis i samme rekkefølge, men..

U3

Det må kunne kodes sånn at du finner dataene på rett plass. Men... Jeg vet ikke om det vil gå litt utenfor din jobb...?

E

Ikke nødvendigvis, men som du sa, at det ikke er en.. Klart det kan være tid å spare der, men kanskje ikke veldig mye?

U3

Nei.

E

Som du sa, at det er en 8-10 forskjellige verdier å taste inn.

U3

Ja, det som tar mest tid er den jobben å databehandle metall ?? ion konsentrasjonene fra dette instrumentet. Der vet jeg at de på geovitenskap som eier instrumentet har laget seg en macro da, som automatiserer det mye lettere. Hvis vi kan komme inn på det senere kan jeg jo vise deg hvordan excel filen ser ut i utgangspunktet, så kan du jo se om det er mulig å gjøre noe der.

E

Ja.

U3

Og så har du jo den hovedbiten her av informasjon om implantatkomponentene, som er inndelt på hoftedelen og på kopp-delen i altså ??, i bekkenet. Og disse dataene har jo da <Admin user> etterhvert kunne importere fra registeret.

E

Ja, fra disse SPSS filene?

U3

Ja. I begynnelsen så tastet vi inn disse manuelt. Den biten har blitt redigert mange ganger, først var det veldig enkelt, det bare stod ja det er hofte og sånn og sånn. Type proteser. Men nå har du stem-name og katalognummer der, produsent, størrelse på stammen, hvilket materiale den er laget av og så videre. Så det er blitt ganske detaljert. Men der må du nok snakke med <Admin user> om detaljene, hva som kan forbedres der.

E

Ja, jeg har snakket såvidt med han om det, han sa det var noen begrensninger på at det var litt miksing og triksing hvis disse feltene var lenger enn sånn og sånn, men akkurat den biten er det lettest å ta med han som faktisk har laget det.

U3

Ja. Og så har vi en boks for vevsanalyse, som jeg også da er ansvarlig for. Der vi på en måte begynner makroskopisk, hvordan ser vevsprøvene ut. Sånn som den ligger, med egne øyne, sånn makroskopisk beskrivelse. Fyller inn den boksen. Så krysser vi om vi har mikroskopi-bilder. Først så er det da vanlig optisk mikroskopi av vevssnitt, krysser av, og da har vi disse bildene, tatt med litt forskjellig mikroskop, så har du undermapper her. Det må jo da og lastes opp manuelt. Det gjør jeg, og det gjør <User 1>. Stort sett. Og så en annen forsker som også tar litt sånne bilder. Og så kan man gå enda mer ned i oppløselighet, detalj. Og se på elektronmikroskopiske bilder. Og Trannisjonsmikropiske bilder. Og da har vi laget masse bokser her, som vi kanskje ikke er helt happy med. Noen for partikkelstørrelse, form, lengde, bredde og om den inneholder elementene jern, krom, kobolt og så videre.

E

Mm. De dataene der, hvor kommer de fra? Der man faktisk observerer... kommer det fra et apparat det og?

U3

Ja. Spesielt denne røntgenanalysen gir kvantitative data ut. Men de er ikke importerbare, så hittil har vi bare sagt at vi fant titan, så vi sier "yes" i der. Hvis vi ikke fant noe så skulle vi kanskje skrevet "no", men vi har latt den stå tom. Så kan vi skrive eventuelt andre element som ikke står på listen nederst. Så den er veldig kvalitativ da. Eh... har vi noe kvantitativt her? Her står det particle size under en viss størrelse. Og så har vi et mellomrom og over en viss størrelse. Akkurat den biten er vi kanskje ikke helt happy med. Så her er det forbedringspotensiale.

E

Ok. På.... på hva som...?

U3

På imaging-data generelt. Ehm..

E

Når dere trykker inn disse verdiene her? Hva er utgangspunktet da? Er det en Excel-fil igjen? Eller er det direkte fra et instrument?

U3

Nei, det er rett og slett en utskrift fra instrumentet, med en graf og bilde. Det er et foto og et type bilde som forteller om elementsammensetningen. I denne nedre delen her. Som har en analysedel. Her nede er det stort sett bare kvalitative data. Som bilder og en beskrivelse av det vi ser på bildene. Her har jeg skrevet histologisk "almost the entire tissue full with black particles. few ?? particles was observed. " Og så en annen teknikk - "particles seem to have sharp edges, not round. " Det er mer sånn beskrivelse, kvalitativ beskrivelse. Men i tillegg til det så kan vi ha måledata, der vi gjør en bildeanalyse. Vi har et programvare som jeg ikke har brukt her, der vi kan telle partiklene, og du kan måle størrelsen på partiklene. Det får du ut i for eksempel en excel-fil. Men der og har vi tatt og plukket det sånn selektivt, altså tatt og skrevet det inn her manuelt i etterkant. Litt på samme måten som de metallnivåene i blod.

E

Er det noen konkrete funksjoner du savner i systemet slik det fremstår nå?

U3

Eh... Ja...

E

Altså noe... som du...

U3

Nei, kommer ikke på noe i farten nå...

E

Nei, det er jo forsvåvidt en god ting det.

U3

Ja, men jeg er sikker på at det er noe.

E

Ja, noe som rett og slett ikke er der da.

U3

Ja. Nei, vi har klart å dytte inn det meste av analyser vi gjør, men det kunne som sagt vært organisert enda bedre layoutmessig og importeringsmessig. Så... men der krever det å gjøre en jobb før det kommer inn i databasen, altså organisering av data. Så, slik at det blir mulig å importere, som sagt.

E

Dette systemet, eller excel-arket for knær, har du det slik at du kan åpne det her?

U3

Ja, har jeg... Jeg har aldri, tror jeg, brukt det.

E

Du har ikke brukt den direkte selv?

U3

.. Jeg regner med at den ligger på samme plass på serveren. Nei, jeg har kanskje tatt en kort kikk. Men der ville jeg ha snakket med hun <User 2> som sitter med det akkurat nå. Du skal snakke med henne også, sant?

E

Ja, imorgen.

U3

Så hun har jo sikkert litt innspill på den. Først så lurte vi jo på om vi skulle bare ha en database, om det var mulig å integrere begge deler i en. Men så så vi det at siden komponentene er så forskjellige, så... Ja du kunne jo hatt bare en litt lenger første-ark da, som tok høyde for de komponentene, men da ville det bli mange felt som ville være tomme hvis det var en annen protese.

E

Ja, eller eventuelt at den kun viste de feltene hvis det var sånn, og de feltene hvis det var sånn.

U3

Ja, altså at du først klikket sånn om det var hofte, kne, som åpner da to ulike...

E

Ja, faner eller...

U3

... to ulike grensesnitt. Fordi jeg liker størrelsen på vinduene og. Men det er kanskje litt mye på det første grensesnittet da. Selv om det vi vanligvis fyller inn er disse pasientdataene først, og så analysedelen sist. Denne komponentdelen er mer automatisert. Den blir importert... Men dataene vil jo komme inn på veldig forskjellig tidspunkt. Du klarer nesten aldri å fylle inn en komplett record samme dagen. Det må du være klar over at vi får informasjon veldig sånn stykkevis og delt. Og vi må alltid mase på kirurgen, "kan du ettersende de og de dataene", sånn som røntgenbilder, de kommer alltid på etterskudd. Og vi må snakke med registeret for at de skal sende disse dataene.

E

Så dere kan sitte her et implantat som dere har ikke disse dataene på?

U3

Ja, vi vet ofte veldig lite om de. Derfor måtte vi jo da lage vårt eget skjema som kirurgen fyller ut så mye som mulig. Men i tillegg må de fylle ut leddregisteret sitt skjema, som er enda mer detaljert.

E

Som har noen av det samme da?

U3

Veldig mye overlapp. Så da er det sikkert mange kirurger som blir litt lei av å fylle ut skjemaer. Der de må svare på mye av de samme tingene. Så vi har lyst å lage et veldig enkelt skjema. Der de nærmest bare gir koblingsnøkkelen, at vi tildeler et record-number, som er koblet til pasientens navn, så kan vi få resten fra leddregisteret sine data. Det er det som er planen. For da slipper vi å spørre om de samme tingene. Det er noen ting som er unikt for oss da, der vi ber de skrive litt om vevsprøvene og blodprøvene som de har samlet. Og hvor vevsprøvene er tatt og så videre, der har vi laget en tegning og så krysser de av. Så den biten må vi jo beholde antageligvis.

E

Apropos den henting fra registeret, er det noe som alle gjør, eller er det en som tar seg av det. Å forespørre disse dataene fra registeret, og forsåvidt importere de og?

U3

Det har vært stort sett <Admin user>. Det.. siden han har bygd databasen så har han...

E

Sikkert lettere for han å...?

U3

... tatt den kommunikasjonen der. Slik at han får det på det formatet han vil. Og <User 1>... har gjort det i mindre grad. Mens jeg bruker da stort sett BioBank-ansvarlige som min kontaktperson. Og jeg har alltid

brukt han for at han går til leddregisteret, han er medlem der, så han kan da av og til sende dataene til oss. Så jeg har av og til bedt om data...

E

Er det fordi det går raskere da eller er det fordi det er greiest for deg siden du har kontakt med han?

U3

Ja, det er fordi han er min nærmeste kontaktperson og at vi jobber på de samme prosjektene. Men i summa summarum er det jo noen på leddregisteret som må hente ut dataene og sende en pakke. Av og til, og i begynnelsen, så fikk vi jo bare kopi av leddregister-skjemaet og så la vi det inn, i den versjonen vi hadde da. Men det så vi jo da var ganske tungvint, og ikke noen fremtidig løsning. Så derfor har <Admin user> prøvd å bearbeide de til å heller gi oss tilgang, kanskje ikke direkte tilgang til deres database, men via sånne delfiler som de lager til oss. Der vi lister de pasientene vi trenger data på. Det må vi vel uansett, vi vil jo ikke ha data på pasienter vi ikke har i vårt system, det er jo helt unødvendig. Så vi har jo ikke sånn grådig mange... 94 records her da.

E

På hofter?

U3

Ja, på hofter. Nei, det... 176 record numbers. Men det er noen som... noen som er ledig på veien da.

E

Ok. Er det fordi ting har blitt fjernet eller...?

U3

Nei, det er fordi vi tildelte intervall...

E

Til studiene..?

U3

Ja. De 20 første hadde den og den inklusjonskriterien. Og de 20 neste. Derfor er det en god del records som ikke er samlet enda.

E

Fordi dere ikke fikk tak i mange nok som var innenfor kriteriene?

U3

Ja, ja. Så derfor så står det noen tomme på veien. Men ifra, tror jeg, hundre eller noe, har det gått stigende. En pluss en. Og det vil det gjøre videre. Så frem til 100 så er det mange hull, og så etter 100 så går det stigende.

E

Så det vil ikke bli... så hvis det er en ny studie så vil det ikke bli satt av på samme måte som..?

U3

Nei, for eksempel knærne er samlet fra 1. Så vi kaller det knee 1, 2, 3 og så videre. I begynnelsen så hadde jeg mitt eget system der jeg kalte titanproteser for ett nummer og andre protesene på et annet nummer. Så da har jeg en boks her da som heter studienavn, som da er det gamle studienummeret. Så det eksisterer jo ikke da for den siste recorden. Men på den.. den første jeg samlet heter HUS1, altså Haukeland sykehus nummer en. Så var det noen som heter Elv fra Elverum og så... Dia for Diakonhjemmet. Altså sykehusrelatert. Så vi valgte å beholde den siden mange av rørene var merket på den gamle måten. Så om vi skal ta det gamle systemet med oss videre må vi diskutere. Her står også registeret og nummer for denne casen. Det er et sekssifret nummer.

E

Ok, så det er bare en identifiseringsnøkkel i registeret.

U3

Ja.

E

Bruker dere den til noe internt eller er den mer hvis dere skal ha noe mer...?

U3

Den er da når vi skal importere fra de, så er det greit å referere til det nummeret. Jeg tror også det står på de papirskjemaene de bruker. Og de har jo et par sekretærer da som taster inn alle dataene fra alle skjemaene.

E

Men når dere snakker om disse her internt så er det disse record numberene dere refererer til eller?

U3

Ja, da er det dette record numberet som vi bruker. Det er det som er nøkkelen til alt for oss.

E

Da regner jeg med og at i et nytt system vil det være et poeng å beholde den rekkefølgen på de som er der nå?

U3

Den må være fast, ja. Kan ikke begynne å re-kode de. De nummerne er også kodet til en kodeliste som han BioBank-ansvarlige har, der en kolonne inneholder det nummeret koblet da til pasientens navn og personnummer, som ikke vi på laben har lov å lagre. Bare i en sånn kort periode der vi driver og avkoder, til vi har fått alle opplysningene fra kirurgen. For kirurgen han er nødt til å vite personnummeret. Så da er det en fordel at vi beholder det til vi har fått komplementert alle dataene. Og det kan ta noen uker. Når vi har fått alle dataene så leverer vi da de pasientsensitive dataene til han <BioBank administrator>, så lagrer han de...

E

Og de er i papirform?

U3

Ja, papir. Så har vi en perm der vi lister alle da pasientene, kodeboken kan vi si.

E

Så dere slipper å ta hensyn til de sensitive dataene inni selve systemet?

U3

Ja. Vi har ikke lov, av etiske grunner, å merke noe av våre data med personnummer eller navn på pasientene. Av og til kan vi kanskje bruke initialene, så lenge det ikke er sporbart. Så... vi har jo en etisk godkjenning fra den lokale komiteen her da. Rek, kaller de det. Rek-vest. Til å bruke dataene i studier som er begrenset til da det vi har søkt om. Vi kan ikke bruke det til hva som helst. Vi kan søke på nytt og utvide bruken av materialet.

E

Men søker dere for hvert prosjekt da eller er det en sånn...?

U3

Nei, så lenge vi passer inn under hovedprosjektet, og det har de stort sett gjort hittil, som jeg startet når jeg var postdoktor, så er det en ganske vid godkjenning. Og den involverer alle de målingene som er involvert her, men for eksempel ikke gen-analyse, det har jeg ikke søkt om. Så hvis vi skulle gjøre det, så måtte jeg ha søkt om en utvidelse av rettighetene. Og vi søkte jo EU ifjor sammen med mange andre land, som hadde lyst å gen-karakterisere vevsprøver og blodprøver. For fremtidige pasienter. Da måtte vi ha søkt etisk komité

om å gjøre det. Men vi ville nok ikke gjort det fordi vi allerede har samlet ?? og jeg tror ikke det materialet er i rette kvaliteten. De er på en måte blitt destruert, det er lagret på et sånt... ja, på formalin og et sånt lagringsmedium som jeg tror tar knekken på den gen-infoen.

Appendix C Interview transcript - Lab users

Wednesday September 4th kl 14:30 - Haukeland university hospital, lab room

E: Eirik Berntsen

U1: User 1

U2: User 2

E

Ok, so first of all some background information. What's your position here at the lab, your job title?

U1

Engineer. I'm here at this lab, I also do testing, material testing.

E

Do the results from that testing end up in the database as well?

U1

Ehm...

E

No?

U1

No, not in the... no. But maybe it was a good idea?

E

Well, I'm not sure what kind of data you collect from that.

U1

No... We test a lot of bone cement and... we also... I haven't thought about the results in a database.

E (to U2)

And you, what's your position here at the lab?

U2

I'm a postdoc, and... My main task is now this new retrieval study. To see... to find some answers why these implants have failed, so I will use the database.

E

Have you started using the knee data... excel-spreadsheet?

U2

Yes, I have started just to summarize all the information I have so far.

E

So inputting data in the knee database, that's also your task?

U2

Mmm. Mm.

E

So you'll use that data or...

U2

Also for example we have the x-rays for our patients. So I will measure a lot of angles and stuff like this. Measurements. And I think that this should also be stored in the database. And I just have an excel-sheet right now.

E

But you have looked into this hip-database, and that works kind of what you would like the knee database to work or?

U2

I don't know, I haven't really looked. I just you... I have also one button for x-rays so that's similar yeah.

E

From what <Admin user> told me the type of data that's to be stored in the knee database is really mostly the same that is being stored in that database. So you don't really use the knee database at all?

U2

No.

E (to U1)

Okay. And how often do you use the hip database? Is it weekly, daily..? Or is it just when you get new...

U1

Yeah, yeah. Only when I get new implants.

E

Okay.

U1

Yeah.

E

And how often would you say that is?

U1

Eh...

E

It probably varies a lot?

U1

Yeah, it varies a lot, but maybe twice a month? Yeah, not more than two.

E (to U2)

Mm. And you with the knee data, how often do you use that information? Daily or weekly or...?

U2

Right now? So the excel-sheet you mean?

E

Yes.

U2

Ah... Every second day I put something there.

E (to U1)

Ok, so fairly often. How well do you feel that the hip database is supporting your work right now?

U1

Yeah, it has been this kind of way to get it perfect. It's... It's not...

E

Yes, because you've been here a long time haven't you?

U1

Yes. It's not perfect yet, cause this... alternatives you have in the database is different from the alternatives that the doctors write down. So it would be more kind of similar to... yeah, all our forms should be kind of....

E

Standardized?

U1

Yeah, sound the same.

E

What's the solution when you...

U1

Because they use these medical terms that I'm not into.

E

What happens when that happens, is the field left empty or...?

U1

Yeah, I have to ask "Is this the same as this? » Yeah, so that would be nice if it was kind of yeah, if the doctor had a form they could fill in with tick-off, and then the same kind of words or alternatives. In the database. Like reason for failure, can vary a lot.

U2

That's also ... different from the... different from the registry and the questionnaire. Different answers. So you don't know which one to trust.

U1

So if everyone would have the alternatives it would help a lot.

E

And do you see any other areas that have potential for improvements from your perspective?

U1

Ehm.... It's not a big problem, it was in the beginning, because when I put in this record database, it didn't automatically come in the other, I had to get to the right number and make a new one and then it came, so it wasn't a big problem, but in the beginning when I wasn't so familiar with it, it was like "where is it, haven't you put it in?!".

E

So you've learned your way around the problem?

U1

Yes, yes. It's not a big deal.

U2

I haven't worked with this, but is it possible to take the data and do like statistics, to extract the data or to...? I don't know. Then it would be nice to....

E

Yeah, you can export it to SPSS or...

Appendix D User testing task instructions

TEST INSTRUCTIONS

TASKS:

Task 1)

Create and save a new record with the following information:

Surgeon: Tore Thorsen

Record date: Todays date

Date of birth: 16.05.1954

Weight: 75

Height: 180

Additional info: Test record 1

Radiographs: Yes

Tibial baseplate: Yes

Tibial insert: Yes

Femur: No

Patella: Yes

Blood samples: 1

Tissue samples: 2

Cement samples: 2

Synovial fluid: No

Implant removed: 14.12.2013

Task 2)

Find the date of birth of record 5

Task 3)

Find the blood sample values of record 7

Task 4)

Find the tissue sample values of record 9

Task 5)

Edit the weight of record 4

Task 6)

Find the product name of the femur cement component of record 9's second revision.

Task 7)

What is the status of record 11?

Task 8)

Make a query that finds all records that have Tore Thorsen as the surgeon

Task 9)

Make a query that finds all records that have Jan Jensen as the surgeon AND has "Smerter" as a removal reason

Appendix E Nielsen's usability heuristics

Visibility of system status:

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world:

The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom:

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards:

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention:

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Recognition rather than recall:

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use:

Accelerators—unseen by the novice user—may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design:

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors:

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation:

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

References

- Abrahamsson, P., & Warsta, J. (2003). New directions on agile methods: a comparative analysis. *Software Engineering, 2003. Proceedings. 25th International Conference on* (pp. 244-254). Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1201204
- Agarwal, R., & Umphress, D. (2008, March). Extreme programming for a single person team. In *Proceedings of the 46th annual southeast regional conference on xx - acm-se 46* (p. 82). New York, New York, USA: ACM Press. Retrieved from <http://dl.acm.org/citation.cfm?id=1593105.1593127> doi: 10.1145/1593105.1593127
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001). *Agile Manifesto* (Vol. 9). Retrieved from <http://agilemanifesto.org/>
- Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry, 189, 194.*
- Budde, R., Kautz, K., Kuhlenkamp, K., & Züllighoven, H. (1992). *Prototyping*. Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from <http://www.springerlink.com/index/10.1007/978-3-642-76820-0> doi: 10.1007/978-3-642-76820-0
- Bugnion, L. (2013). *MVVM Light Toolkit*. Retrieved 26/02/14, from <http://www.galasoft.ch/mvvm/>
- Cimino, J. J., Socratous, S. A., & Clayton, P. D. (1995, September). Internet as Clinical Information System: Application Development Using the World Wide Web. *Journal of the American Medical Informatics Association, 2*(5), 273-284. Retrieved from <http://jamia.bmj.com/content/2/5/273.abstract> doi: 10.1136/jamia.1995.96073829
- Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Addison-Wesley Professional.
- Conallen, J. (1999, October). Modeling Web application architectures with UML. *Communications of the ACM, 42*(10), 63-70. Retrieved from http://dl.acm.org/ft_gateway.cfm?id=317677&type=html doi: 10.1145/317665.317677
- Coronel, C., Morris, S., & Rob, P. (2012). *Database Systems Design, Implemen-*

- tation, and Management, 10th Edition.* Course Technology Ptr.
- Doyle, B., & Lopes, C. V. (2008, January). *Survey of Technologies for Web Application Development*. Retrieved from <http://arxiv.org/abs/0801.2618>
- Fog Creek Software. (2014). Trello. *Trello.com*. Retrieved from <https://trello.com/>
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional. Retrieved from <http://www.amazon.com/Design-Patterns-Elements-Reusable-Object-Oriented/dp/0201633612>
- Git-scm.com. (2014). Git. *Git-scm.com*. Retrieved from <http://git-scm.com/>
- Gossman, J. (2005). *Introduction to Model/View/ViewModel pattern for building WPF apps*. Retrieved 20/02/14, from <http://blogs.msdn.com/b/johngossman/archive/2005/10/08/478683.aspx>
- Gossman, J. (2006). *Advantages and disadvantages of M-V-VM*. Retrieved 21/02/14, from <http://blogs.msdn.com/b/johngossman/archive/2006/03/04/543695.aspx>
- Haukeland University Hospital. (2013). *The Norwegian Arthroplasty Register*. Retrieved 02/05/13, from <http://nrlweb.ihelse.net/eng/default.htm>
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004, March). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105. Retrieved from <http://dl.acm.org/citation.cfm?id=2017212.2017217>
- Humphrey, W. (1996, May). Using a defined and measured Personal Software Process. *IEEE Software*, 13(3), 77–88. Retrieved from http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=493023 doi: 10.1109/52.493023
- IBM. (2014, April). SPSS software. *IBM.com*. Retrieved from <http://www-01.ibm.com/software/no/analytics/spss/>
- Jazayeri, M. (2007, May). Some Trends in Web Application Development. In *Future of software engineering (fose '07)* (pp. 199–213). IEEE. Retrieved from <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4221621> doi: 10.1109/FOSE.2007.26
- Jeffries, R. (2009). *Extreme Programming adventures in C#*. O'Reilly Media, Inc.
- JetBrains. (2014). Resharper. *Jetbrains.com*. Retrieved from <http://www.jetbrains.com/resharper/>
- K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler,

- J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mallor, Ken Shwaber, J. S. (1999). Embracing change with extreme programming. *Computer*, 32(10), 70–77. Retrieved from http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=796139 doi: 10.1109/2.796139
- Khan, O. M. A. (2012). Implementing Repository Pattern with Entity Framework — Architects Zone. *Architects Zone*. Retrieved from <http://architects.dzone.com/articles/implementing-repository>
- Kosch, H., & Döller, M. (2005). Multimedia database systems: Where are we now? *Special Session Talk to be given at the IASTED DBA-Conference in Innsbruck*. Retrieved from <http://www.dsc.ufcg.edu.br/~baptista/cursos/BDCopin/MMDBoverview.pdf>
- Krasner, G., & Pope, S. (1988). A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object oriented programming*. Retrieved from http://www.itu.dk/courses/VOP/E2005/VOP2005E/8/_mvc/_krasner/_and/_pope.pdf
- Lim, R. (2011). Navigation with MVVM. *Rachel Lim's Blog*. Retrieved from <http://rachel53461.wordpress.com/2011/12/18/navigation-with-mvvm-2/>
- Long, M., & Rack, H. (1998, September). Titanium alloys in total joint replacementa materials science perspective. *Biomaterials*, 19(18), 1621–1639. Retrieved from [http://dx.doi.org/10.1016/S0142-9612\(97\)00146-4](http://dx.doi.org/10.1016/S0142-9612(97)00146-4) doi: 10.1016/S0142-9612(97)00146-4
- March, S. T., & Smith, G. F. (1995, December). Design and natural science research on information technology. *Decision Support Systems*, 15(4), 251–266. Retrieved from [http://dx.doi.org/10.1016/0167-9236\(94\)00041-2](http://dx.doi.org/10.1016/0167-9236(94)00041-2) doi: 10.1016/0167-9236(94)00041-2
- McCauley, R. (2001, December). Agile development methods poised to upset status quo. *ACM SIGCSE Bulletin*, 33(4), 14. Retrieved from <http://dl.acm.org/citation.cfm?id=572139.572150> doi: 10.1145/572139.572150
- Microsoft Corporation. (2012). Building Testable Windows Phone Applications. *Microsoft Developer Network*. Retrieved from <http://msdn.microsoft.com/en-us/library/hh830877.aspx>
- Microsoft Corporation. (2014a). Introduction to WPF. *Microsoft Developer*

- Network*. Retrieved from [http://msdn.microsoft.com/en-us/library/aa970268\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/aa970268(v=vs.110).aspx)
- Microsoft Corporation. (2014b). Microsoft Access - database software and applications. *Microsoft.com*. Retrieved from <http://office.microsoft.com/en-us/access/>
- Microsoft Corporation. (2014c). Microsoft Excel spreadsheet software. *Microsoft.com*. Retrieved from <http://office.microsoft.com/en-us/excel/>
- Microsoft Corporation. (2014d). Overview of the .NET Framework. *Microsoft Developer Network*. Retrieved from <http://msdn.microsoft.com/en-us/library/zw4w595w.aspx>
- Microsoft Corporation. (2014e). SQL Server Editions-Express. *Microsoft.com*. Retrieved from <http://www.microsoft.com/en-us/server-cloud/products/sql-server-editions/sql-server-express.aspx#fbid=WNKSfR8HfqZ>
- Microsoft Corporation. (2014f). SQL Server Management Studio. *Microsoft Developer Network*. Retrieved from <http://msdn.microsoft.com/en-us/library/hh213248.aspx>
- Microsoft Corporation. (2014g). TFS Overview. *Visualstudio.com*. Retrieved from <http://www.visualstudio.com/en-us/products/tfs-overview-vs.aspx>
- Microsoft Corporation. (2014h). Visual Studio. *Visualstudio.com*. Retrieved from <http://www.visualstudio.com/>
- Morgan, L., & Finnegan, P. (2007). How perceptions of open source software influence adoption: An exploratory study. *Proceedings of the 15th European Conference on Information Systems (ECIS 2007)*. Retrieved from <http://60.88.dyn.lse.ac.uk/asp/aspecis/20070050.pdf>
- Müller, H., & Deserno, T. (2011). Content-based medical image retrieval. *Biomedical Image Processing*. Retrieved from <http://www.springerlink.com/index/G43LW05254J254U4.pdf>
- Navathe, S., & Elmasri, R. (2010). *Fundamentals of Database Systems*. Pearson Education.
- Nielsen, J. (1995). *Severity Ratings for Usability Problems*. Retrieved from <http://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/>
- Nielsen, J., & Molich, R. (1990, March). Heuristic evaluation of user interfaces. In *Proceedings of the sigchi conference on human factors in computing systems*

- empowering people - chi '90* (pp. 249–256). New York, New York, USA: ACM Press. Retrieved from <http://dl.acm.org/citation.cfm?id=97243.97281> doi: 10.1145/97243.97281
- Noort, R. (1987, November). Titanium: The implant material of today. *Journal of Materials Science*, 22(11), 3801–3811. Retrieved from <http://link.springer.com/10.1007/BF01133326> doi: 10.1007/BF01133326
- Nunamaker, J., & Chen, M. (1990). Systems development in information systems research. In *Twenty-third annual hawaii international conference on system sciences* (Vol. iii, pp. 631–640). IEEE Comput. Soc. Press. Retrieved from <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=205401> doi: 10.1109/HICSS.1990.205401
- Oates, B. J. (2005). *Researching Information Systems and Computing* (Vol. 2005). SAGE Publications Ltd.
- Olan, M. (2003, December). Unit testing: test early, test often. *Journal of Computing Sciences in Colleges*, 19(2), 319–328. Retrieved from <http://dl.acm.org/citation.cfm?id=948785.948830>
- Oracle Corporation. (2014). MySQL :: The world's most popular open source database. *MySQL.com*. Retrieved from <http://www.mysql.com/>
- Orlikowski, W. J., & Iacono, C. S. (2001, June). Research Commentary: Desperately Seeking the "IT" in IT Research—A Call to Theorizing the IT Artifact. *Information Systems Research*, 12(2), 121–134. Retrieved from <http://isr.journal.informs.org/content/12/2/121.abstract> doi: 10.1287/isre.12.2.121.9700
- Ræder, J. (2012). Er eldrebølgen på vei inn i sykehusene? *Tidsskrift for den Norske Laegeforening*, 132(23).
- Reese, L., & Iberle, K. (2011). Kanban what is it and why should I care? *PNSQC 2011 Proceedings*. Retrieved from http://www.uploads.pnsqlc.org/2011/papers/T-31_Reese_Iberle_paper.pdf
- Royce, W. (1970). Managing the development of large software systems. *proceedings of IEEE WESCON*. Retrieved from http://leadinganswers.typepad.com/leading_answers/files/original_waterfall_paper_winston_royce.pdf
- Rubin, J., & Chisnell, D. (2008). *Handbook of Usability Testing: Howto Plan, Design, and Conduct Effective Tests*. Wi-

- ley. Retrieved from <http://www.amazon.com/Handbook-Usability-Testing-Conduct-Effective/dp/0470185481>
- Schwaber, K., & Sutherland, J. (2013). Scrum Guide. *Scrum.org*. Retrieved from <https://www.scrum.org/Scrum-Guide>
- Shan, T., & Hua, W. (2006). Taxonomy of Java Web Application Frameworks. In *2006 IEEE International Conference on e-Business Engineering (ICEBE'06)* (pp. 378–385). IEEE. Retrieved from <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4031677> doi: 10.1109/ICEBE.2006.98
- Shen, M., Rong, G., & Shao, D. (2013). Integrating PSP with agile process: a systematic review.. Retrieved from http://www.atlantis-press.com/php/download_paper.php?id=5660
- Smith, J. (2009). *THE MODEL-VIEW-VIEWMODEL (MVVM) DESIGN PATTERN FOR WPF*. Retrieved 26/02/14, from <http://msdn.microsoft.com/en-us/magazine/dd419663.aspx>
- StatensHelsetilsyn. (1999). *Scenario 2030 - Sykdomsutviklingen for eldre fram til 2030*. Retrieved 05/03/13, from http://www.helsetilsynet.no/upload/Publikasjoner/utredningsserien/2030_sykdomsutvikling_eldre_ik-2696.pdf
- TechSmith. (2014). Usability Testing Software. *Techsmith.com*. Retrieved from <http://www.techsmith.com/morae.html>
- Travis, D. (2007). Heuristic Evaluation with Morae. *UserFocus.com*. Retrieved from <http://www.userfocus.co.uk/articles/morae-he.html>
- Tsichritzis, D., & Klug, A. (1978). The ANSI/X3/SPARC DBMS framework report of the study group on database management systems. *Information systems, 3(3), 173-191.*
- U.S. National Institutes of Health. (2013). *ClinicalTrials.gov*. Retrieved 10/05/13, from <http://clinicaltrials.gov/ct2/home>
- Virzi, R. A. (1992, August). Refining the Test Phase of Usability Evaluation: How Many Subjects Is Enough? *Human Factors: The Journal of the Human Factors and Ergonomics Society, 34(4), 457–468*. Retrieved from <http://hfs.sagepub.com/content/34/4/457.short> doi: 10.1177/001872089203400407