

Development of a Flexible Help Function for Increased Use and Value for Web Applications

Kristina Hansson
Veronica Johansson
Kerstin Lyngfelt

EXAMENSARBETE

Högskolan Trollhättan · Uddevalla
Institutionen för Informatik och Matematik

Uppsats för filosofie kandidat i Datavetenskap

Development of a Flexible Help Function for Increased Use and Value for Web Applications

Kristina Hansson
Veronica Johansson
Kerstin Lyngfelt

Examinator:
Lektor Lena Pareto
Handledare:
Linn Gustavsson

Institutionen för Informatik och Matematik
Institutionen för Informatik och Matematik

Trollhättan, 2003
2003:D23

DEGREE PROJECT

Development of a Flexible Help Function for Increased Use and Value for Web Applications

Kristina Hansson
Veronica Johansson
Kerstin Lyngfelt

Abstract

Web applications of today are often advanced and it is not obvious to the user how to use them and which functions they provide. This creates a need for some kind of help system that can provide help and support to the user.

Users differ when it comes to experiences in handling different applications, which creates a need for help systems that support users in various levels and situations. It should also provide more than one method for the user to find the information needed. Help systems like that exist in different forms and we have chosen to develop a help function that is easy to adjust to different web applications and different users needs.

By thinking in the user's perspective, and by taking part of some of the users view, we have investigated the existing needs and what solutions that are preferred. In order to meet all different needs our help function is designed as an online manual including hierarchical structured, indexed and searchable information.

To maximize the use of a help system it should be adjusted to the application it is meant to support. Applications are continuously modified and developed and therefore the help system should be easy to update and maintain in order to keep the help system up to date.

Our help function belongs to Empir AB's product, *Virtuella Kontoret* but is developed to be easily adjusted to other similar applications. The help function is built on a database model in order to be adjustable to different applications and to various users needs.

Besides giving the users a tool for help and support in the use of the web application the help function also contains a graphic tool for the administrator. With this tool the administrator can change, delete and add new information in the help function and through this keep it well adjusted to its users needs.

| | | | |
|-------------------|--|------------------|-------------|
| Publisher: | University of Trollhättan · Uddevalla, Department of Informatics and Mathematics Box 957, S-461 29 Trollhättan, SWEDEN Phone: + 46 520 47 50 00 Fax: + 46 520 47 50 99 | | |
| Examiner: | Lektor Lena Pareto | | |
| Advisor: | Linn Gustavsson, University of Trollhättan - Uddevalla | | |
| Subject: | Computer science | Language: | English |
| Number: | 2003:D23 | Date: | May20, 2003 |
| Keywords | help system, help function, user centred, web application, help | | |

EXAMENSARBETE

Utvecklingen av en flexibel hjälpfunktion för ökad nytta och värde för webbapplikationer

Kristina Hansson
Veronica Johansson
Kerstin Lyngfelt

Sammanfattning

Dagens webbapplikationer är ofta avancerade och det är inte alltid helt självklart för användaren hur dessa ska användas och vilka funktioner de har. Detta gör att behovet av ett hjälpsystem finns för att ge denna information som stöd och hjälp åt användaren.

När det gäller erfarenheter i att hantera olika applikationer skiljer sig olika användare åt, vilket gör att hjälpsystemet bör kunna ge hjälp och stöd på olika nivåer och i olika situationer. Dessutom bör det erbjuda flera metoder för användaren för att finna den information denne söker för att passa deras skiftande behov och erfarenheter. Hjälpsystem som motsvarar detta finns i olika former och vi har valt att utveckla en hjälpfunktion som lätt ska kunna anpassas till olika webbapplikationer och till olika användares behov.

Genom att tänka ur användarens perspektiv och att ta del av dennes åsikter har vi kartlagt vilka behov som finns samt vilka lösningar som är att föredra. För att möta dessa olika behov är vår hjälpfunktion utformad som en online manual med både hierarkiskt strukturerad, indexerad och sökbar information.

För att användaren ska ha maximal nytta av ett hjälpsystem bör det vara välanpassat för den applikation det hör till. Applikationer kan förändras och utvecklas och för att hjälpsystemet inte ska bli inaktuellt bör det kunna uppdateras och följa med i utvecklingen. Detta når man genom att underlätta underhållet och administrationen av hjälpsystemet.

Vår hjälpfunktion hör till Empir AB:s produkt, *Virtuella Kontoret* men kan justeras för att användas till andra liknande applikationer. Hjälpfunktionen bygger på en databasmodell som lätt kan anpassas till olika applikationer och till olika användares behov. Även designen och utformningen i övrigt är utvecklad för att vara anpassningsbar.

| | | | |
|--------------------|---|---------------|-------------|
| Utgivare: | Högskolan Trollhättan · Uddevalla, Institutionen för Informatik och Matematik Box 957, 461 29 Trollhättan Tel: 0520-47 50 00 Fax: 0520-47 50 99 | | |
| Examinator: | Lena Pareto, HTU | | |
| Handledare: | Linn Gustavsson, HTU | | |
| Huvudämne: | Datavetenskap | Språk: | Engelska |
| Nivå: | Fördjupningsnivå 1 | Poäng: | 10 |
| Rapportnr: | 2003:D23 | Datum: | 20 maj 2003 |
| Nyckelord: | hjälpsystem, hjälpfunktion, användarcentrering, webbapplikation, hjälp | | |

EXAMENSARBETE

Förutom att ge användarna ett verktyg för att finna hjälp och stöd i användandet av webbapplikationen har vår hjälpfunktion ett grafiskt administrationsverktyg. Med hjälp av detta kan administratören ändra, ta bort och lägga in ny information i hjälpfunktionen och därigenom hålla den välanpassad till sina användares behov.

| | | | |
|--------------------|---|---------------|-------------|
| Utgivare: | Högskolan Trollhättan · Uddevalla, Institutionen för Informatik och Matematik Box 957, 461 29 Trollhättan Tel: 0520-47 50 00 Fax: 0520-47 50 99 | | |
| Examinator: | Lena Pareto, HTU | | |
| Handledare: | Linn Gustavsson, HTU | | |
| Huvudämne: | Datavetenskap | Språk: | Engelska |
| Nivå: | Fördjupningsnivå 1 | Poäng: | 10 |
| Rapportnr: | 2003:D23 | Datum: | 20 maj 2003 |
| Nyckelord: | hjälpsystem, hjälpfunktion, användarcentrering, webbapplikation, hjälp | | |

Preface

Division of labour

In this project we are equally responsible and all of us have participated in all parts of the project. For practical reasons we have had head responsibility of some parts each:

Kristina Hansson – Database design, programming

Kerstin Lyngfelt – Programming, ASP .Net

Veronica Johansson – Survey, programming

In the report we wrote separately at first and then rewrote it together. In this way all of us are participants in the whole report.

Acknowledgement

A project like this is both a fun and educational task. At some points even small obstacles on the way takes too long time and too much energy to solve. In these moments we have had a few persons that have helped us and they deserve some of the credit for this job:

Erik Bergman, Empir AB has replied fast and provided us with code suggestions and tips on solutions to our problems. Big Thank You!

Lena Pareto, HTU has guided us through the scientific way of writing an exam report. We still like the red pen!

Linn Gustavsson, HTU who has been a support both in writing the report and in developing the help function. You have been a great help!

Christian Jiresjö, HTU has helped us with installations and troubleshooting in the initial state. We really appreciate your help!

We also like to thank our circle of friends who has helped us and supported us both in the most difficult times and in the more exiting times when everything is working as they are supposed to. The project period would have been harder without you!

Content

| | |
|--|-----|
| Abstract | ii |
| Sammanfattning | iii |
| Preface | v |
| 1 Introduction..... | 1 |
| 2 Background..... | 3 |
| 2.1 <i>Different help and support to users</i> | 3 |
| 2.2 <i>The use of groupware</i> | 3 |
| 2.3 <i>Description of Virtuella Kontoret</i> | 3 |
| 2.3.1 <i>The main areas for the users</i> | 4 |
| 3 Preconditions | 5 |
| 3.1 <i>Software</i> | 5 |
| 3.2 <i>Name standards</i> | 5 |
| 3.3 <i>Demarcations</i> | 5 |
| 3.4 <i>Future development possibilities</i> | 5 |
| 4 Goal..... | 5 |
| 4.1 <i>Increased usability</i> | 6 |
| 4.2 <i>Increased flexibility</i> | 6 |
| 4.3 <i>Increased value</i> | 6 |
| 5 Method..... | 7 |
| 5.1 <i>Analysis</i> | 7 |
| 5.2 <i>Requirements</i> | 8 |
| 5.3 <i>Design</i> | 8 |
| 5.4 <i>Prototype</i> | 8 |
| 5.5 <i>Implementation</i> | 9 |
| 5.6 <i>Evaluation</i> | 9 |
| 5.7 <i>Validation and verification</i> | 10 |
| 5.8 <i>Planning the activities</i> | 10 |
| 5.9 <i>Risk analysis</i> | 10 |
| 6 Development work..... | 10 |
| 6.1 <i>Analysis</i> | 10 |
| 6.1.1 <i>Virtuella Kontoret</i> | 11 |
| 6.1.2 <i>Survey</i> | 11 |
| 6.1.2.1 <i>Frequently used functions</i> | 12 |
| 6.1.2.2 <i>Grading of the functions</i> | 12 |
| 6.1.3 <i>System design</i> | 14 |
| 6.1.4 <i>Analysis of the database</i> | 14 |
| 6.2 <i>Requirements of the system</i> | 16 |
| 6.3 <i>Design</i> | 16 |
| 6.3.1 <i>User interface</i> | 17 |
| 6.3.2 <i>Administratörsformuläret</i> | 18 |
| 6.3.3 <i>Design of the database</i> | 19 |
| 6.3.3.1 <i>Table tblHelpTopic</i> | 19 |
| 6.3.3.2 <i>Table tblHelpText</i> | 19 |
| 6.3.3.3 <i>Table lnkHELPTOPIC_HELPTOPIC</i> | 19 |

*Development of a Flexible Help Function for Increased
Use and Value for Web Applications*

| | | |
|---------|--|----|
| 6.3.3.4 | Table lnkHELPTOPIC_HELPTEXT | 20 |
| 6.4 | <i>Implementation</i> | 20 |
| 6.4.1 | Prototyping..... | 20 |
| 6.4.2 | Writing the code to the program | 21 |
| 6.5 | <i>Evaluation and verification</i> | 21 |
| 7 | Future work..... | 21 |
| 7.1 | <i>Adaptation to .NET</i> | 22 |
| 7.1.1 | Why..... | 22 |
| 7.1.2 | How..... | 22 |
| 7.1.3 | Pros and cons | 22 |
| 7.1.4 | This help function | 23 |
| 8 | Conclusion | 24 |
| 9 | References..... | 26 |

Appendix

| | | |
|------------|--|----|
| Appendix A | – Requirements specification (in Swedish) | 28 |
| Appendix B | – Project plan (in Swedish) | 29 |
| Appendix C | – Gantt-schema (in Swedish) | 30 |
| Appendix D | – Program flow (both in English and Swedish)..... | 32 |
| Appendix E | – Prototypes (in English)..... | 34 |
| Appendix F | – Additional question to survey (in Swedish) | 36 |

1 Introduction

When an application or a system reaches a certain limit of complexity it is not obvious how it can be used. With a help system connected to the application the users can get help and support with using the entire potential of the application. It will also prevent the application from being used in the wrong way. The users can also use the help system to decrease the time for learning the application.

A help system can be differently developed and provided to the users in several ways. One way for a user to get help and information about an application or system is to use a written manual. Another way is to use an online help system, which may be connected to the application or placed on the Internet.

In many online help systems the user gets the opportunity to search for the actual help topic in a hierarchical structure or in alphabetical order. Several help functions also include methods where the user can search for information with keywords or get a guide that leads them step-by-step through a task. When the user has found the desired information it is presented within the help function. This kind of online help system is described later on in this report.

When a help function is developed and implemented for an application it is good practise for the developer to assume that both experienced and less experienced users uses the application. A flexible help system should therefore provide help to the users in different levels and according to their different needs. These aspects are important to consider when a help function is developed according to Ian Sommerville [1].

To accomplish this flexibility many help systems has a user interface that provides different methods for the user to find information. With a database containing the information, the help system easily can provide a lot of information in different levels, which is also easy to update. When a help system consist of a user interface and a database, it can easily be adapted to different applications.

When users have a help system that supports them in using a complex application the usability of the application can increase. This because the users easier can find the methods they need and information about handling the application. The potential of the application can also be more visible for the user. When the user can find information about the application the time learning it can decrease. The usability and value of an application can be measured by how easy it is to learn and use.

Many companies, schools and other groups of today do work organized in groups and may have use of an application to support them in communication, information sharing and document handling. *Virtuella Kontoret* is an example of a groupware application, which is used as a virtual office supporting its users with these tasks.

Empir AB, an IT-consulting company in Uddevalla, has developed *Virtuella Kontoret* and their customers can buy or lease the product. *Virtuella Kontoret* does not have an online help function today to support and help the users in their work, and Empir AB want to offer this service to their customers. Beside Empir AB, the users and administrators are the main stakeholders that will have interests and use of the help function.

With a help function connected to *Virtuella Kontoret* the users can be supported in finding the potentials of it and how to use it in the planned way. They can also use the help function to learn the application easier.

The help function will provide an administrating tool which can be used for keeping the information in the database well suited for the application. Moreover, the administrators can use the help function to find help about this administration tasks and other help as a user.

When *Virtuella Kontoret* contains a help function it will become a more complete and hopefully more useful application than it is today. Since *Virtuella Kontoret* is one of Empir AB products they can use this as a sales argument.

2 Background

Virtuella Kontoret has no help function today but since it can be considered as a complex application it may need one.

2.1 Different help and support to users

People often need a description or a manual to guide them while using a technical product. Their task can be everything from building common furniture to handle difficult software. In our case we will study how software users can get support and help with their software.

One way for the user to get help and support is to read a paper manual that describes the product. Another way is to ask friends and colleagues or to search for information in books, magazines or on the Internet. As a third way the user can find information in an electronic manual or help system which can be published on the Internet or connected to the software itself.

This different help systems have both advantages and disadvantage that have to be analysed before a decision about which one is the best solution in the actual case is taken. This different help systems will be investigated and analysed later in this report to get information about which system is best for web applications like *Virtuella Kontoret*.

2.2 The use of groupware

A groupware is an application that supports people working in groups with communication, document handling and coordinating their common tasks. A groupware works as a workspace for the group in which they can solve their common tasks.

The work in working groups is more and more supported by computer systems. Most applications gives a user-to-system interaction even in a multi user system, but a groupware handle the interaction user-to-user both directly an indirectly, according to Ronald M Baecker [2].

2.3 Description of Virtuella Kontoret

Virtuella Kontoret is used as a virtual office that supports people in work groups with document handling, information sharing and communication. It has five main areas; Discussions, Calendar, Documents, Contacts and a dynamic list with actual events.

Development of a Flexible Help Function for Increased Use and Value for Web Applications

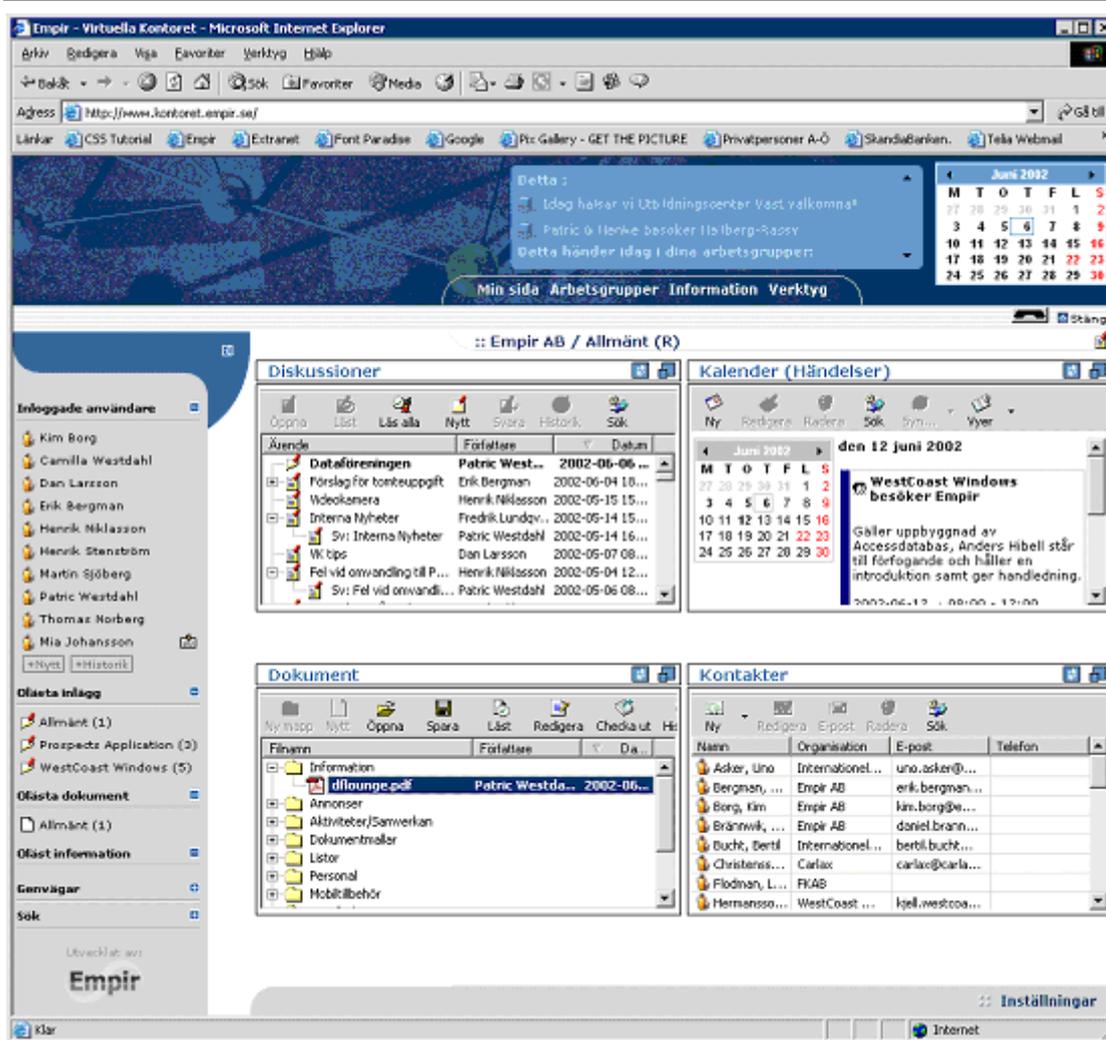


Figure 1 : Virtuela Kontoret © Copyright 2002 Empir AB (modified)[3]

2.3.1 The main areas for the users

In Documents, group members can store, read and work with shared documents. When more than one person works with the same document the built in version handling provide good help. Users can find a documents history in a log where it is shown, for example, who has read the document and when changes may have occurred.

Discussion is the user object in which group members can read contributions from other members in the workgroup. For keeping information about important meetings and other date related information for the user or the group they can use Calendar. The group members can use Contacts for keeping addresses and other information they want for the project work.

The user interface has a dynamic online list with user defined links and links to unread documents and discussions. The name of the users that is logged into *Virtuela Kontoret* at the present becomes links that can be used to send instant messages.

3 Preconditions

Since our help function is meant to be used with *Virtuella Kontoret* there are a few considerations that have to be made.

3.1 Software

Virtuella Kontoret is based on Microsoft's products such as Active Server Pages for script code language and SQL server 2000 as database manager. For maximal compatibility, the help function should be developed in the same languages. Empir AB wants to have the different modules of *Virtuella Kontoret* written in the same standard to simplify the maintenance.

3.2 Name standards

Virtuella Kontoret is using a special name standard. To simplify the future development of the application and the understanding of the code, we are using the same standard. This means that all files related to the help function have the prefix `hlp`. Variable names have prefixes related to their type, for example `intNumber` for an integer.

3.3 Demarcations

Within the project, a tool for inserting text into the system will be provided, but we will not formulate the actual help texts. We will not perform the actual integration of the help function into the application.

3.4 Future development possibilities

Since Microsoft has chosen to develop ASP .NET as part of the .NET framework instead of adding more functionality into regular ASP, it is likely to believe that tomorrow's web applications will be created in ASP .NET instead of ASP. Therefore we will describe how an ASP-application can be adjusted to simplify migration to ASP .NET in the future.

4 Goal

Our goal is to develop a help function that shall serve the users with help text and provide a tool for administration and maintenance of the function. It will be flexible and increase the usability and value for the web application it is connected to.

We want to visualize the possibilities in the application for the user and to support users in learning and using it.

We will develop a database that contains the information and data that the help function requires for presenting help and support for the user.

We will develop a tool for the administrator to update the help function and its database. Since a well-updated database gives more relevant information and hence gives the user a more useful help, it is important that the maintenance can be managed in an easy way.

Both the education of the user and writing the documentation is included in the implementation, according to Nilsson and Pettersson [4]. We will therefore provide Empir AB with documentation about the functionality included in the help function.

4.1 Increased usability

We want to increase the usability of the web application by develop a tool that provides help when and where the user needs it. This tool will also contain a few guides that can be used in initial learning of the application by providing step-by-step instructions.

We will develop a graphic tool for adding, updating and deleting text into the help functions database, which will simplify the administration. If updating and maintenance are easy to perform, it will decrease the maintenance time and the help function will probably be updated more often than if it is cumbersome.

To be able to sell a complex application a good marketing argument is including a help function.

4.2 Increased flexibility

In order to increase the flexibility we will provide different methods of finding information since users have various customs and experiences.

The help function will have a database where all the information in the help function is stored to simplify information access.

The graphic tool will make it possible for the administrator to be located anywhere and through the interface make immediate changes of the content in the database.

By developing a general help function it will suite different versions of *Virtuella Kontoret* and also other web applications. Only a few adjustments will be needed to make the database and the corresponding script files working with similar applications.

4.3 Increased value

The help function will make it possible to make large and complex applications more easy to manage for the user. In many cases, the users find that an application have a greater value if it is useful to them. If the help function increase the usability of the application it increases the value for the user. We believe we will show that our help function increases the usability.

Through the administrator tool updating is simplified, which saves time for the administrators and that increases the help functions value.

As the application develops and more modules are added, the company will have more arguments when promoting the product and will be able to charge more for it. This will increase the economical value of the web application for the company.

5 Method

In this project we chose to work with a user-centred approach because the help function is developed to serve the user. This development process can contain different phases and the ones we have used will be described later in this part.

To make an application for the users that fit their needs the developer have better chances to succeed if the application is developed from the users view from the beginning, according to Sommerville [1]. In a user centred process the developer should make prototypes that are tested by the users and from the user's perspective. In every step of the prototyping it is tested from the user's perspective and the next step is taken in the direction to make the product more suited for the users. The user interface will come first and the technical solutions will be connected to it.

To get the needed information from the users we will use a survey made by Linn Gustavsson [5]. Users that use *Virtuella Kontoret* today have participated in this survey. We will also use *Virtuella Kontoret* for our development work to get personal experience from using the system. The administrator view we will get from Empir AB because they administrate *Virtuella Kontoret* today.

When actually program the application we will use programming books [7,8,9,10], that gives us information on how an ASP application could be programmed and how to design a functional database. We will also search for this kind of information on websites [12,13,14] since information written in books is to be considered static while websites can have the latest news on the subject.

5.1 Analysis

During the analysis we will investigate the user's need of help and support. This will be done through a survey that gives us their opinion about difficulties in *Virtuella Kontoret* and our own experiences.

We will also interview administrators to get knowledge about their needs regarding finding help and administrating the help function. Interviews will also be our method of getting information from Empir AB about their needs in the help function.

We will also investigate *Virtuella Kontoret* and the help functions of other similar applications to find different ways to solve the same task. A help function can use different methods to let the user find the information and to read it.

5.2 Requirements

During the analysis we will try get the stakeholders needs concerning the help function. To be able to develop a product that fills these needs we have to decide which requirements the help function should have. The requirements from the different groups vary and will be handled accordingly.

Users have their requirements of how to find help and how to read it. They also have requirements about the content of the information in the help function. The user interface will be created to meet the user's requirements of finding information. The administrator tool will help the administrator to provide the proper information in the database of the help function.

All requirements should be well thought through and correspond to the users needs regarding the help function. The requirements are the foundation of the whole development process, and will therefore affect the quality of the final product.

5.3 Design

When we make the design of the help function we will find a solution, which realizes the requirements. The design of the two interfaces should aim at making it easy for the users and administrators to find the methods that they want to use for the actual task.

The design of the technical solution and code will be designed to support the user and administrator interfaces with their tasks. The help function is built on a database solution that has a design, which best suits, this version of the help function but is prepared for adaptations to other web applications.

5.4 Prototype

According to Ian Sommerville [1] prototyping can start with a simple prototype that only includes the most important requirements. As the work proceeds the prototypes are developed step by step into the final product. With a prototype it is easier to show the stakeholders what the application would be like.

Our first prototypes will be paper examples of the user interface and the administrator interface. These prototypes will be used to find a good design and to find connections between the interfaces and the technical solution. These prototypes will also be used as examples in discussions with users and administrators.

Next step is to make more functional prototypes where we can develop and test function by function until the interfaces reaches the functionality that fill the requirements. With a test database the functions that require data from a database can be developed independent of the other. This will end up with the same result but will make it possible to develop different parts simultaneous without interference.

A test interface can be used when developing the database and be used until the prototypes of user and administrator interfaces reaches the level of development so they can be connected to the database. This makes the database development more independent of the development of the rest of the help function.

Since our method is evolutionary prototyping, every working function is implemented and tested together with the latest version of the prototype. This makes each prototype a new version with a higher level of functionality until the requirements are filled and the final version is developed.

5.5 Implementation

When using a prototype the methods required in the function are implemented gradually. Each implemented method is tested from the users' perspective and measured by the requirements.

According to Gulliksen and Göransson [6] is it important to let the user take an important part of the development work at an early stage. It is also important to realize that the users are not a homogeneous group. Since *Virtuella Kontoret* is a new application to us, we can get our own user perspective from using *Virtuella Kontoret* in our development work. We have also used the earlier mentioned survey [5] to get a general opinion from the users about *Virtuella Kontoret*.

A central part in user centred development is the usability, according to Nilsson and Pettersson [4]. To get information about which help might be useful in *Virtuella Kontoret* we will have discussions with the developers of *Virtuella Kontoret*.

5.6 Evaluation

According to Nilsson and Pettersson [4] there are four main parts that are tested during an evolutionary usability test. First the software is tested if it supports the user in their actual task. The next test is if the function is user friendly regarding accessibility consistency of how the user interaction with the system. The user acceptance is tested to find the motivation for the user to use the system. At last the users competence to handle and understand the system and its functions.

Since we will use prototypes in our development the evaluation and testing will be done for each new prototype in the development. We will make this test from the user's perspective to see if the function is working as intended. When it comes to how the help function supports the users in their work and how user friendly the function is, we will test it from the information we will get from administrators of *Virtuella Kontoret*. We cannot do the last two tests about user acceptance and competence because of our demarcations to not write the help texts or connect the help function to *Virtuella Kontoret*.

5.7 Validation and verification

Using evolutionary developed prototypes means that every method in the help function will be tested when it is developed. When two or more functions are developed and working satisfactory, they are tested together. For each loop in the prototyping the prototype is tested and checked if the product works the way intended and if it fills the requirements. Finally when we get to the last prototype the validation and verification is done.

This is at the theoretical level but we will work with the validation and verification in this way to minimize the risks for mistakes. Every implemented method has its aim and task, and we will test and check the method according to that.

5.8 Planning the activities

To organize and follow the development process we planned and documented our time and activities. This project documentation can be found in the appendix as an example of a work process when a help function is developed.

5.9 Risk analysis

Even though this is a relatively small project we wanted to perform a risk analysis to establish our cooperation methods and to be prepared if unexpected situations occur. Experiences from other projects have taught us that if risks are predicted and risk handling are thought of, the risks for major problems decreases.

The risk analysis is documented as tables placed in the appendix. They can be read as an example of which risks that can jeopardize a small project like ours and how they can be handled.

6 Development work

We have worked user centred and iterative through the use of prototyping.

6.1 Analysis

To find out if a help function was needed or not for *Virtuella Kontoret* we started this project with a study of *Virtuella Kontoret*. This gave us the impression that a help function can be of great use for the users because *Virtuella Kontoret* is a complex application, which is not obvious how it should be handled.

Since a big amount of information has to be stored in the help function it will be as complex that it need a database to store the information. This database has to be flexible to fit into *Virtuella Kontoret* today and easily adjusted to fit into a more developed

version because applications seldom are static. Later in this report the database will be more described.

When a user is working with an application, problems often occur when trying to understand the product. It can happen to users that are both experienced or not and therefore it is important to develop a help function that serves users at different levels. By knowing what the users need to get help with, we can prepare the help function after the most common problems.

We used the result from a survey developed by Linn Gustavsson [5], to get information from the users of *Virtuella Kontoret*. The users find it difficult to understand some functions and they do not even know that some exists. The survey also helped us to find out what functions of *Virtuella Kontoret* that are most frequently used. This is more described later in this report.

In discussions with the administrators of *Virtuella Kontoret* and developer at Empir AB, we realized that a tool consisting of a graphic interface built on a database solution when administrating the help function was to prefer rather than working in for example SQL. A tool like this simplifies updates and maintenance since the administrator does not have to be located where the database is and the tool makes sure that all tables in the database are filled with the right information. The administrators also need a tool to search for information and help about the administration tasks.

6.1.1 Virtuella Kontoret

Virtuella Kontoret is a groupware with different main parts that supports the users in their work in groups. These main parts handle for example communication and document handling and this makes *Virtuella Kontoret* a complex application that is not obvious for the users. Since some functions are easy to find and use but some are not, a help function can make *Virtuella Kontoret* more easy to use.

Both experienced and non-experienced users use *Virtuella Kontoret* today. They have different needs for help with *Virtuella Kontoret* regarding both the methods to find the information and the content of it.

The non-experienced user can have use of a step-by-step manual that guides the user from the beginning to the end of the task. They may prefer to find the information they search for with help of a hierarchical tree structure that is a model of how to find the topics in *Virtuella Kontoret*. The more experienced user can use the search method if they need help with more advanced parts in *Virtuella Kontoret*.

6.1.2 Survey

To get information about what kind of help and information the users need we took part of an existing survey [5]. Participants of the survey were students at University of Trollhättan/Uddevalla who used *Virtuella Kontoret* during a course.

Even though the survey consisted of a number of questions we focused on the two most relevant to us, and the result is presented below.

6.1.2.1 Frequently used functions

Which function do you use most frequently?

| Function | First week | After four weeks |
|--------------------------------------|-------------------|-------------------------|
| Read information from teachers | 6 | 7 |
| Upload files | 3 | 7 |
| Read new documents | 11 | 3 |
| Download file | | 1 |
| Use the shortcut to unread documents | 1 | 1 |
| Participate in discussions | | 1 |
| All functions | | 1 |

This was an open-ended question that not everybody in the survey answered. The numbers in the table are the amount of people who use that function most frequently.

6.1.2.2 Grading of the functions

The users have graded which features are most common and they have also commented which parts in the application that might be difficult to understand.

There are 12 proposed functions in the question used for evaluation by grading. 1 works very badly, 2 works quite bad, 3 works acceptably, 4 works quite well and 5 works very well.

| | |
|---|---|
| 1 | Listed logged users in the left menu. |
| 2 | Send messages to logged in users |
| 3 | Able to plan the time in the calendar |
| 4 | Able to upload files |
| 5 | The system keeps a record of the version of the files the system has uploaded |
| 6 | Able to share files with others |
| 7 | Able to discuss with others in the workgroup |

Development of a Flexible Help Function for Increased Use and Value for Web Applications

| | |
|----|--|
| 8 | Able to keep a record over contacts by saving their personal information |
| 9 | Able to hide the top frame and the frame on the left side |
| 10 | The reminder in the left menu when user has a unread document |
| 11 | Able to put up links to various workgroups |
| 12 | The links in the top menu to “good to know” pages |

This graph illustrates the use of functions during the first week of the study

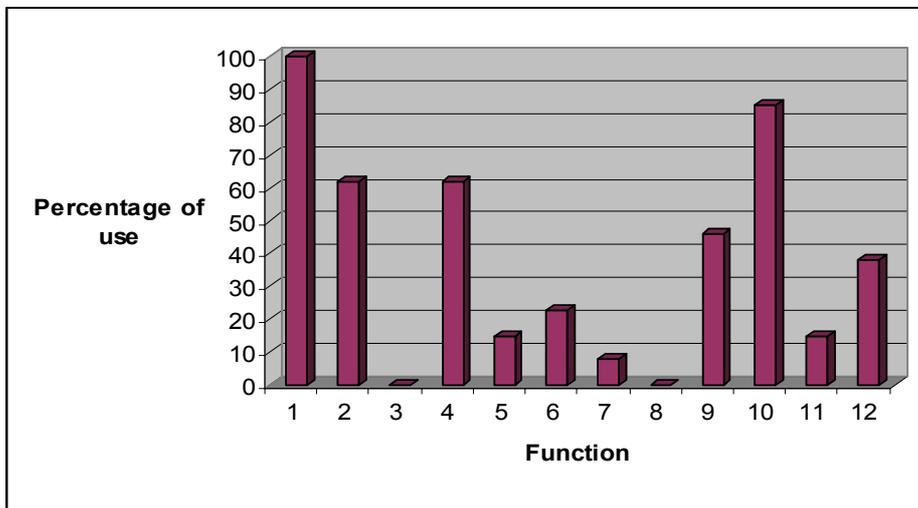


Figure 2 : Use of functions during the first week of study

This graph illustrates the use of functions during the four weeks of the study

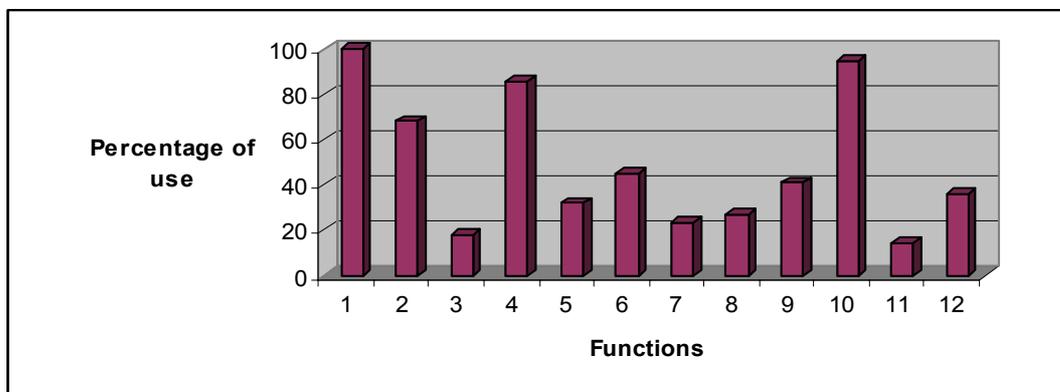


Figure 3 : Use of functions during four weeks of study

In order to know what should be included in the step-by-step guides we made an additional survey in Swedish where we sent questions to users of *Virtuella Kontoret*. The questions we asked are listed in appendix [F] but since we have not received any answers yet we cannot publish any results. We have prepared step-by-step instructions and Empir AB can use the answers in further development of the help function.

6.1.3 System design

To be able to decide what kind of help system that was most suitable for *Virtuella Kontoret*, we studied help functions in existing applications. Since *Virtuella Kontoret* is developed as a Windows application we concentrated on these. We tested and tried several functions to get an idea of how they are constructed.

We found out that many software products on the market contain a help function that supports the users with information and help topics for the application. The user can search for a specific problem, and get a guide to help solving the problem. Another possibility is to have all topics presented in alphabetical order or hierarchical like in a tree structure. The help functions could be used in parallel with the application it supports so the user can find help while using the application.

We chose to make the help function as similar to other Windows applications as possible because some of *Virtuella Kontorets* features only work in a Windows environment. Another reason is that users can easier adapt to a familiar system, than to learn a new one.

6.1.4 Analysis of the database

The database will serve the help function with both data that will be presented to the users and parameters used administratively. By studying the existing database for *Virtuella Kontoret* we got an idea of how to name the tables and attributes. All tables have the prefix tbl, and where link tables are needed there prefix are lnk. The attributes will have prefixes that reflect their type for example intValue is an integer.

Virtuella Kontoret is supporting many languages, and so will the help function. It will also have the functionality to relate one topic to another and to use permission levels to decide whether help texts should be visible to users or not.

*Development of a Flexible Help Function for Increased
Use and Value for Web Applications*

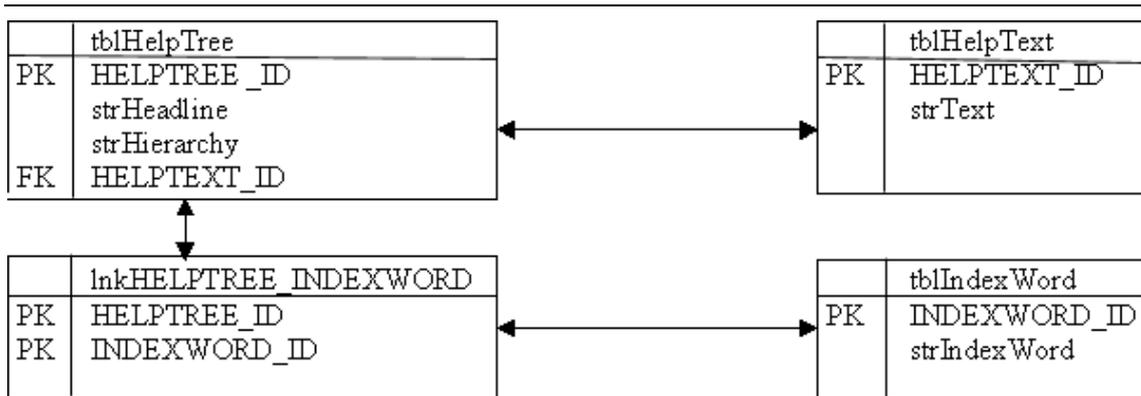


Figure 4 : The first database model

Our first design choice was to use special keywords when searching in the database and to store these keywords in a separate table. This design would have been effective and fast while searching, but harder to find the right keywords to make the search accurate. In discussions with the administrator at Empir AB we decided that even if it is a bit slower to search for words in the entire texts, this option would be preferred if and provide more accurate search results for the user.

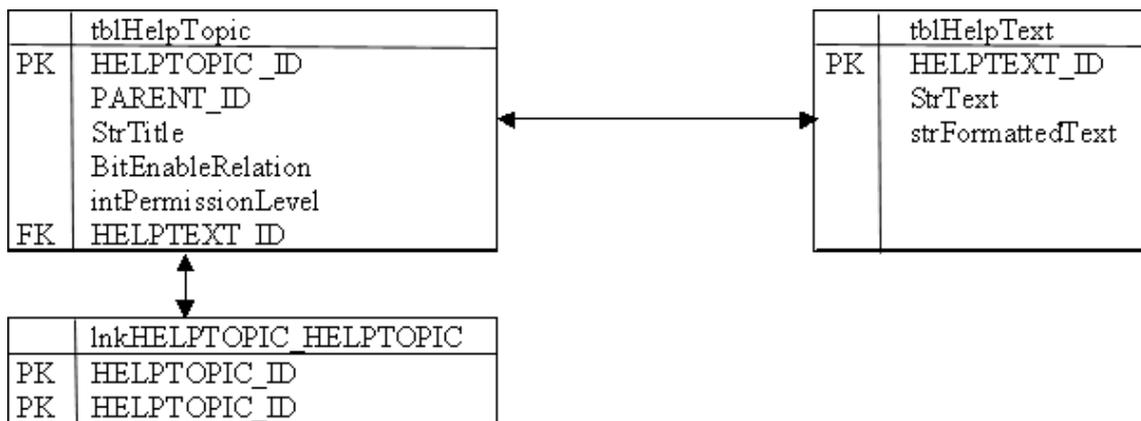


Figure 5: The second database model

The next design choice was with only two tables, one containing all data about each topic and one link table to be able to relate one topic to another. This choice would be easy to program, but not so functional in the long run. Since we want the help function to support more than one language the same topic would have been stored as many time as the languages it supports. At the present this would not have been a problem but since we want this help function to be flexible and used for a longer period we created a new model.

have been specified in parameters. By doing so, it is only necessary to alter one variable to change for example the coloring to fit each company's demands.

When installing the help function all that is necessary is to run an SQL script that creates the tables in the database and then run a start script that loads the database tables with default information. After that the administrator can alter the tables through the graphic interface.

6.3.1 User interface



Figure 7 : The user interface

Since users are used to look for help in different ways it is good to provide them with more than one way to find information. The most common ways are hierarchically like tree structures, alphabetical lists or by searching for keywords. A flexible help function should provide all these possibilities to suite most kinds of users.

To provide all these possibilities, a menu was needed to switch between the different kinds of help methods. To provide this we could either create layers in the HTML code or link different pages to the menu.

Layers would provide less waiting time when switching between the help options, but would consume more loading time. All functions would be located in the same page, which would have been very long and hard to troubleshoot if a change has to be done.

Instead we chose to use an additional frameset with buttons at the top frame, which are linked to the different help pages in the frame beneath. This choice are faster to load, but may take some time when switching between pages. It simplifies maintenance, since it is easier to change or replace a complete page than to change a part of a long code

page. Most users prefer to find information in the hierarchical tree structure and therefore that page is default. When the user clicks on a topic, the related text is immediately presented in the main window.

The next alternative is to show all the topics in alphabetical order that the user can click to get the information.

The last alternative is to search for keywords or a string in the database. As most help systems on the web it is possible to search for words in the text by setting a '+' sign in front of the word if it has to be included in the text or a '-' sign if it is not included in the text. It is also possible to search for a whole string, placed between double quotes, ""

6.3.2 Administrator interface

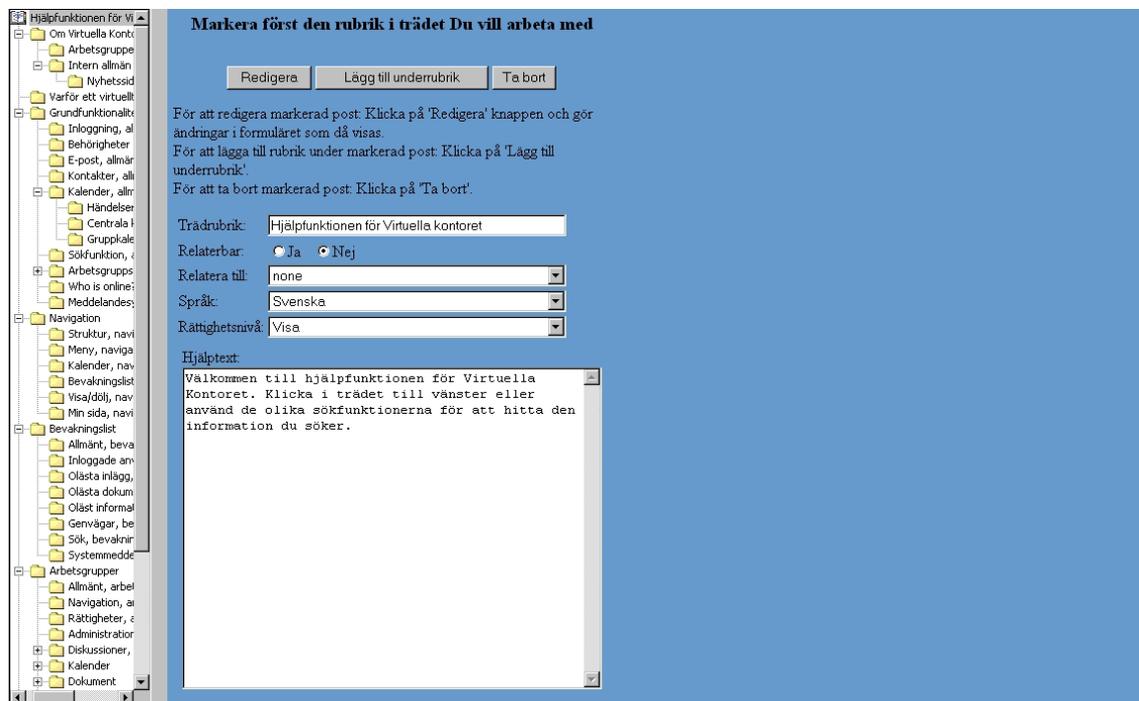


Figure 8: The administrators interface

The tool for administering the content of the function is designed with the same tree structure on the left side as the user interface. With a similar design we want the administrator to see the same view as the user and also see changes in the tree structure immediate after they are made.

When selecting a topic in the tree, the form is filled in with the related data. Then the administrator can either chose to change data, add a new topic to the tree or delete a topic. It will be possible to set permission levels on the information and by that control who is allowed to see different kind of information.

6.3.3 Design of the database

As earlier explained we have decided to create a database containing of two tables of data and two link tables to connect them. The naming of tables and attributes follows the standard of *Virtuella Kontoret*. This means that tables with the prefix *tbl* gets the rest of the name from the table's primary key. Link tables with the *lnk* prefix usually has two primary keys and the name contains both.

6.3.3.1 Table *tblHelpTopic*

The table that contains data about the topics has the following attributes:

| | |
|--------------------|---|
| HELPTOPIC_ID | This attribute is an integer and also the primary key and identifier of the table which is automatically set by the database. |
| intParent | An integer that represents the topics parent in the tree structure. |
| BitEnableRelation | A bit set either to true or false to indicate if the topic could be related to or not. |
| IntPermissionLevel | An integer that represents the permission level of the topic. |
| bitVisible | A bit set to true if the topic will be visible. |

6.3.3.2 Table *tblHelpText*

This table contains the titles and the help texts:

| | |
|--------------|--|
| HELPTTEXT_ID | This attribute is an integer and also the primary key and identifier of the table which is automatically set by the database. |
| StrTitle | This string is the title that is shown in the tree, in the indexed list or in the search results. The size is limited to 50 bytes. |
| txtText | This is the help text that is not limited. The attribute contains a pointer of 16 bits to the location of the whole text. |

6.3.3.3 Table *lnkHELPTOPIC_HELPTOPIC*

This link table connects topics to each other. Both attributes are primary keys and foreign keys to *tblHelpTopic*:

| | |
|---------------|--|
| HELPTOPIC_ID1 | Integer that represents the first related topic. |
| HELPTOPIC_ID2 | Integer that represents the topic the first is related to. |

6.3.3.4 Table *lnkHELPTOPIC_HELPTEXT*

This link table connects the help topic with the related help text. The two first attributes are primary keys and foreign keys to respectively `tblHelpTopic` and `tblHelpText`:

| | |
|--------------|---|
| HELPTOPIC_ID | An integer that represent the help topic. |
| HELPTEXT_ID | An integer that represents the help text. |
| intLanguage | An integer who represents what language the text is written in. |

6.4 Implementation

While developing this help function we used evolutionary prototyping.

6.4.1 Prototyping

We chose to work with evolutionary prototyping [1] because we knew most of the requirements at the beginning of the development. The prototypes were used for testing the system and for validation and verification.

The order of which the methods are implemented is important for the development. The different parts are depending of each other and therefore we made plans of in, which order the prototypes should be made.

The first prototype was a user interface sketch on a paper. With that we tested different methods of finding and presenting the information to the user. In discussions with Empir AB we came to the conclusions that a user interface like the standard interface in the help functions in Windows system is a good design because most users is used to that. This prototype also helped us to find which information that could be shown to the user and to the administrator. Some of this information had to be stored in the database and this prototype helped us to analyze and find the attributes of the database.

The following prototypes included one new major function each time a new one was developed. At first the connection to the database was implemented and then the hierarchical tree that the user could click in.

Since the administration tool inserts information into the database it is an important part of the help function. The user interface with its methods is depending on the database and if it is implemented. For this reasons the next prototypes became the development of the administrator tool and its methods, which was implemented one by one, starting with the insert-method. We finally developed the user interface with its methods step-by-step.

For each step in the evolution of the development the method was tested to see if it solved the task it was intended to solve. It was also tested against the requirements to see if the method filled them.

6.4.2 Writing the code to the program

One way to shorten the developing time for an application is to reuse existing code. It is a waste of time to rewrite code blocks and functions if they already are written. Therefore, when writing the code to the help function, we had access to *Virtuella Kontorets* source code and could see their code structure. Even if there was not so much code we could reuse as it was, we could write the new code in the same structure in order to simplify the system administrator's maintenance of the program. We also had access to Empir AB's ActiveX controls such as the tree structure we used in both the user interface and the administrators tool for maintenance.

Most of the logic is built inside the database, through Stored Procedures, instead of processed in the script code. The reason for this is to gain a more efficient execution of the commands and also to hide the code from unauthorized persons. The only task for the script code is collecting values and sending them into the database. The database then performs the transactions and returns the selected data that the script code requested.

The most difficulties when producing the code was not the overall structural issues, but small details such as correct paths and single/double quotes. These things tend to increase the developing time very much even for experienced programmers.

6.5 Evaluation and verification

According to Ian Sommerville [1], the validation check if the application is suitable for its purpose and the verification is to check if the application matches the specifications.

Since we had developed the specifications and requirements for the help function when we started the design phase we used prototyping for testing the function and not for getting the specifications. When the test was made we also made the validation and verification.

Each prototype is tested and measured with the specifications and with the use of it. The use of it we got from the users through the survey and from the administrators through interviews.

Every method in the administrator tool and the user interface had its purpose and task to solve. In the test we tested if the method we tested solved the task that it was aimed to solve.

7 Future work

Active Server Pages is for the moment Microsoft's leading language in developing web applications due to its flexibility and simplistic support for dynamic web page creation.

Predictions for the future are pointing to an increased number of web applications and distributed services via Internet and for this reason, ASP are no longer sufficient for the increasing demands. To meet these new demands Microsoft has developed a whole new framework called .Net. ASP .Net is a part of this large framework and because of this; they had to drop some of the compatibility to older versions of ASP.

When updating web applications, one aspect to consider is if it should be written in the existing language or in a newer language. In the case of ASP vs. ASP .Net it is possible to let the two languages coexist, but not always without difficulties. Both Wrox's book ASP .NET [7] and Microsoft's web site for developers [11] have a few tips on how to simplify future adaptation from ASP to ASP .Net.

7.1 Adaptation to .NET

When considering adaptation to ASP .Net there is a few things that need to be thought of.

7.1.1 Why

Looking back in history, progress in web design and programming has progressed in high speed and that speaks in favor for using a new developed language that is more adapted to greater demands. Old ASP might decrease the flexibility in the long run.

7.1.2 How

There are some aspects to consider when writing server-side ASP code making a future migration to .Net easier. They are as follows:

In ASP it is possible to use variables without declaration. This is a source for errors that can be hard to locate. The use of the `Option Explicit` statement at the top of the ASP pages can prevent such errors since it enforces variable declaration before use.

Functions should be declared in `<SCRIPT></SCRIPT>` blocks instead of `<% %>` blocks in order to make it more similar to other .Net code. When calling functions in ASP it is possible to only write the function name as for example: `MyFunction`. To force the use of parentheses and to make the code cleaner it is better to use the **call** keyword as for example: `call MyFunction()`.

Other recommendations are not to use default properties and to always close open resources. This is to prevent errors and to have more control of what code that is executed.

7.1.3 Pros and cons

The fast development rate speaks for using the most modern and recent developed languages. Possibilities increases and the languages are adjusted to the computers

capacity of today. Old applications that are written in old languages easily become less modern and are therefore less supported. In strictly economical, short-term perspective, it is best to keep the old language as long as possible. The developers already know the language and syntax and learning a new language takes time. New applications are always instable before being used as intended and tested by real users.

7.1.4 This help function

This help function is written in ASP, mainly because the application it is meant to be migrated with, *Virtuella Kontoret*, is written in that language. When writing the code, the aspects above have been considered when possible, in order to make it easier to migrate to ASP .Net in the future. Some parts of the code are not possible to migrate as they are, and needs to be rewritten to function with ASP .Net.

8 Conclusion

Our goal was to develop a help function, which should serve the users with help and support by different methods. We also wanted the help function to visualize the possibilities in the application and support users in learning and using it.

As we aimed the help function now has three ways for the user to find help texts in the user interface. Users can find topics either from a hierarchical tree structure, from an indexed list in alphabetic order or by searching for words in the texts. We find these possibilities flexible for the user since various methods can serve different users. The help function also contains a few guides to help the user with the most common tasks.

The help function visualizes the possibilities in the application it is connected to because the user can find information about it. We find that the use and value for the application will increase when the user is provided with a tool to find information and support.

We wanted to develop a database that contains the information and data which the help function require.

The help function contains a big amount of data and to keep it actual and easy to update a database can be useful. When developing the database we had to reconsider when we tested the prototypes we realized that new data had to be stored. The final database is prepared for further development of the help function and therefore we can consider it to be flexible in its design.

One of our goals was to make a tool for administration and maintenance of the help function.

The Administration tool has functionality for editing, adding and delete topics and related texts. The tool has direct access to the database through the script files and simplifies maintenance of the help function. The administrator can access the administrator tool from any browser and perform updates in real time. It is possible to change existing texts and to add new topics to the tree structure or indexed list. It is also possible to remove topics permanently from the database. With this tool that simplifies the administration of the database the administrator can keep the database well updated. If the help function contains incorrect information the help function is less useful and valuable for the user. The administrator can use this tool to make adjustments in the content of the database, which makes it easy to adjust the help function to the actual application.

We wanted to provide Empir AB with a module that can be connected to their applications and be used as a good marketing argument.

*Development of a Flexible Help Function for Increased
Use and Value for Web Applications*

We consider the help function being flexible since applications with a help function have a greater value than an application without one. Empir AB is now provided with a help function as a good sales argument.

9 References

- [1] Sommerville, Ian (2001). *Software Engineering*. Harlow: Addison-Wesley
- [2] Baecker, Ronald M (1993). *Readings in Groupware and Computer-supported Cooperative Work*, Morgan Kaufmann Publisher Inc. San Mateo, Calif: Kaufmann, cop.
- [3] Empir AB (2002). *Usermanual* Uddevalla
- [4] Nilsson, Anders G, Pettersson, John Sören (2001). *On methods for systems development in professional organizations*. Lund : Studentlitteratur [x] Kroenke, David (2000). *Database Processing*. Upper Saddle River, N.J.:Prentice Hall, cop.
- [5] Gustavsson, Linn (2002). *Usability of Virtual Office - A case study*. Högskolan Trollhättan Uddevalla
- [6] Gulliksen, Jan, Göransson, Bengt (2002). *Användarcentrerad systemdesign*. Lund : Studentlitteratur
- [7] Anderson et al (2000). *Professional ASP .NET*. Birmingham: Wrox, cop.
- [8] Waymire, Richard (2000). *Learn SQL server 2000 in 21 days*. Indianapolis, Ind: Sams; Hemel Hempstead :Prentice Hall
- [9] Esposito et al. (2000). *Professional ASP DataAccess*. Birmingham: Wrox Press
- [10] Childs, Lomax and Petruscha (2000). *VBScript in a nutshell*. Sevbastopol: O'Reilly

Websites

- [11] <http://msdn.microsoft.com/library> 2003-05-05
- [12] http://w3.one.net/~kevinw/VB/l_ocx.htm 2003-05-05
- [13] <http://samples.gotdotnet.com/quickstart/aspplus/doc/syntaxandsemantics.aspx> 2003-05-05
- [14] <http://www.w3schools.com/asp/> 2003-05-05

Appendix A – Kravspecifikation (Requirements specification)

Gruppdeltagare:

Kerstin Lyngfelt, Kristina Hansson och Veronica Johansson, studenter vid HTU i Trollhättan.

Syfte:

Empirs produkt, Virtuella kontoret (VK), har idag ingen utförlig hjälpfunktion. Genom att utveckla en intuitiv hjälpfunktion underlättas arbetet med VK för både användare och administratörer.

Omfattning:

Hjälpfunktionen kommer att bygga på en databaslösning i MS SQL Server 2000 och gränssnittet utformas i ASP. Databaslösningen skall vara flexibel för att lätt kunna redigeras via ett användargränssnitt. Lösningen skall också ha stöd för att lägga in hjälptexter som kommer att produceras av Empir. Implementeringen av hjälpfunktionen i det existerande VK kommer att utföras av Empir.

Administratörer kommer att ha tillgång till alla hjälptexter i funktionen, medan vanliga användare bara har tillgång till den hjälp de behöver.

Resultat:

En databaslösning med ett grafiskt gränssnitt som skall gå att implementera i VK med tillhörande hjälptext.

Rättigheter:

Vid projekttidens slut överlämnas produkten till Empir i befintligt skick, men projektgruppen behåller rätten att återanvända den egenskrivna koden och lösningen i framtiden.

Tidsplan:

2003-03-17 Officiell projektstart

2003-04-30 Preliminär inlämning av produkten till Empir

2003-05-20 Officiellt projektslut

Appendix B – Projektplan (Project plan)

Bakgrund

Virtuella kontoret är en webbapplikation utvecklad av Empir AB..... I dagsläget saknar *Virtuella kontoret* en hjälpplikation som kan hjälpa användarna.

Mål Inriktning och Avgränsning

Målet är att skapa en hjälpfunktion för *Virtuella Kontoret* med databaskoppling till MSSQL2000.

Mallarna för hjälptexterna kommer att lagras i XML i databasen. Eftersom .NET är ett nytt språk på ingång kommer vi även att beskriva hur denna hjälpfunktion kan portas till .NET.

Implementationen av hjälpfunktionen kommer att hanteras av Empir AB, men vi ska skapa förutsättningarna för den.

Strategi och Metodval

Vi planerar att erbeta efter Vattenfallsmodellen och Spiralmodellen i kombination. Varje fas behandlas för sig där man i slutskedet reflekterar över risker i nästa fas. Vi kommer att summera och analysera föregående fas och sedan planera inför nästa.

Alla deltagare är delaktiga i alla delar av arbetet för att ta tillvara varje individs specifika kunskap och erfarenhet.

Översiktlig aktivitet och tidsplan

Se Gantt schema

Projektbudget

Tillgång till ASP-server SQL-databas på Högskolan i Trollhättan/Uddevalla.

Arbetstiden beräknas vara 9 veckor, med tre dagars bortfall på grund av helgdag. Detta ger 344 timmars arbetstid per deltagare och hela projektet planeras till 1032 timmar.

Projektorganisation

Arbetet kommer att ske jämställt i demokratisk anda. Ingen projektledare kommer utses.

Information Inflytande frågor

Gruppmöten kommer att hållas kontinuerligt för att hålla alla deltagare informerade och uppdaterade.

Möten där uppdragsgivaren sitter med kommer hållas.

Skriftlig dokumentation kommer hållas löpande och kommer att finnas färdigställt då projektet är avslutat.

Förväntad slutprodukt

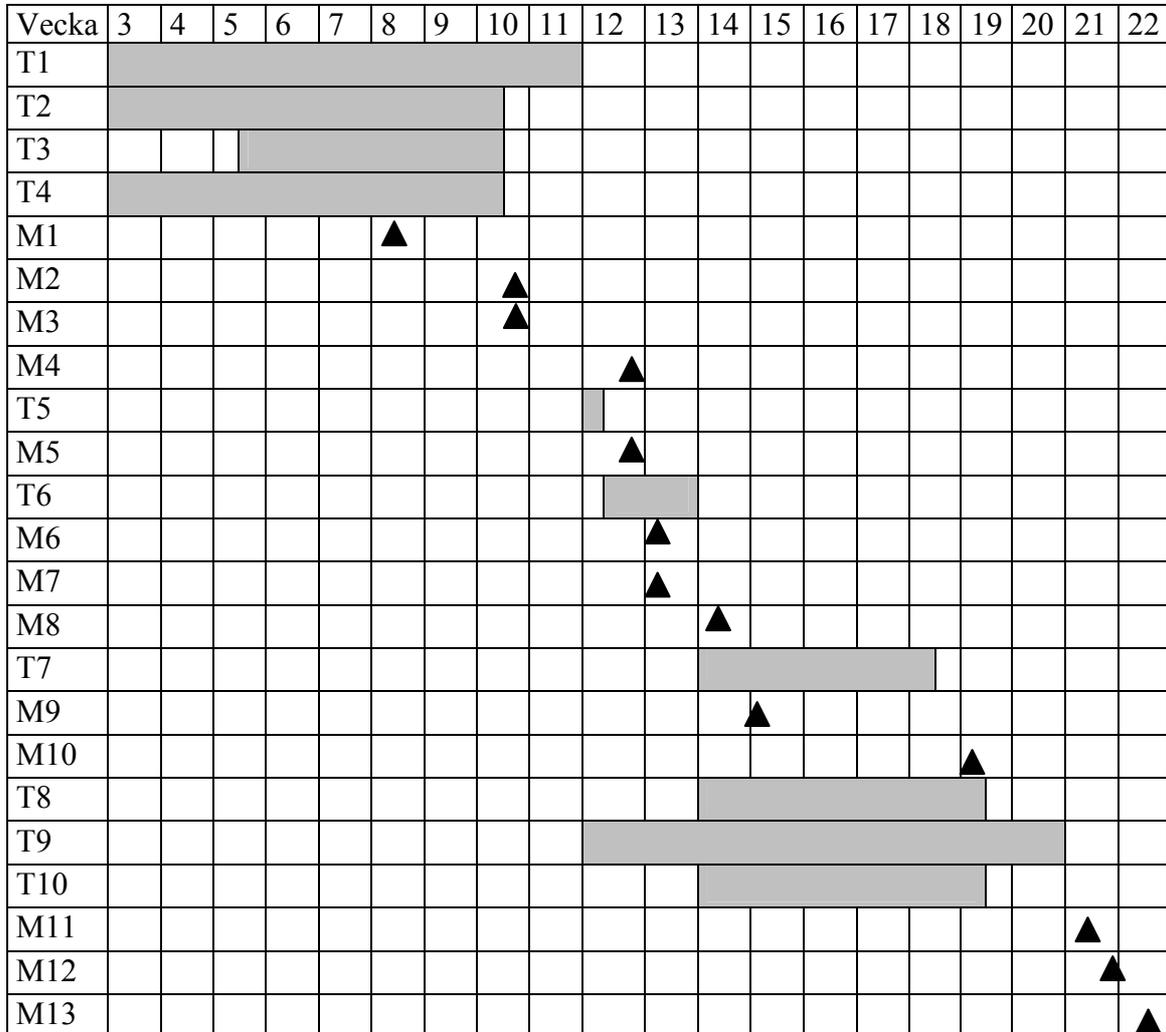
En hjälpfunktion till *Virtuella Kontoret* samt en rapport.

Förväntad sluteffekt

Hjälpfunktionen ska hjälpa användaren att använda funktionen Virtuella Kontoret samt stärka Virtuella Kontorets konkurrenskraft.

Appendix C – Gantt Schema

Gantt schema för arbetet med hjälpfunktionen till Virtuella kontoret



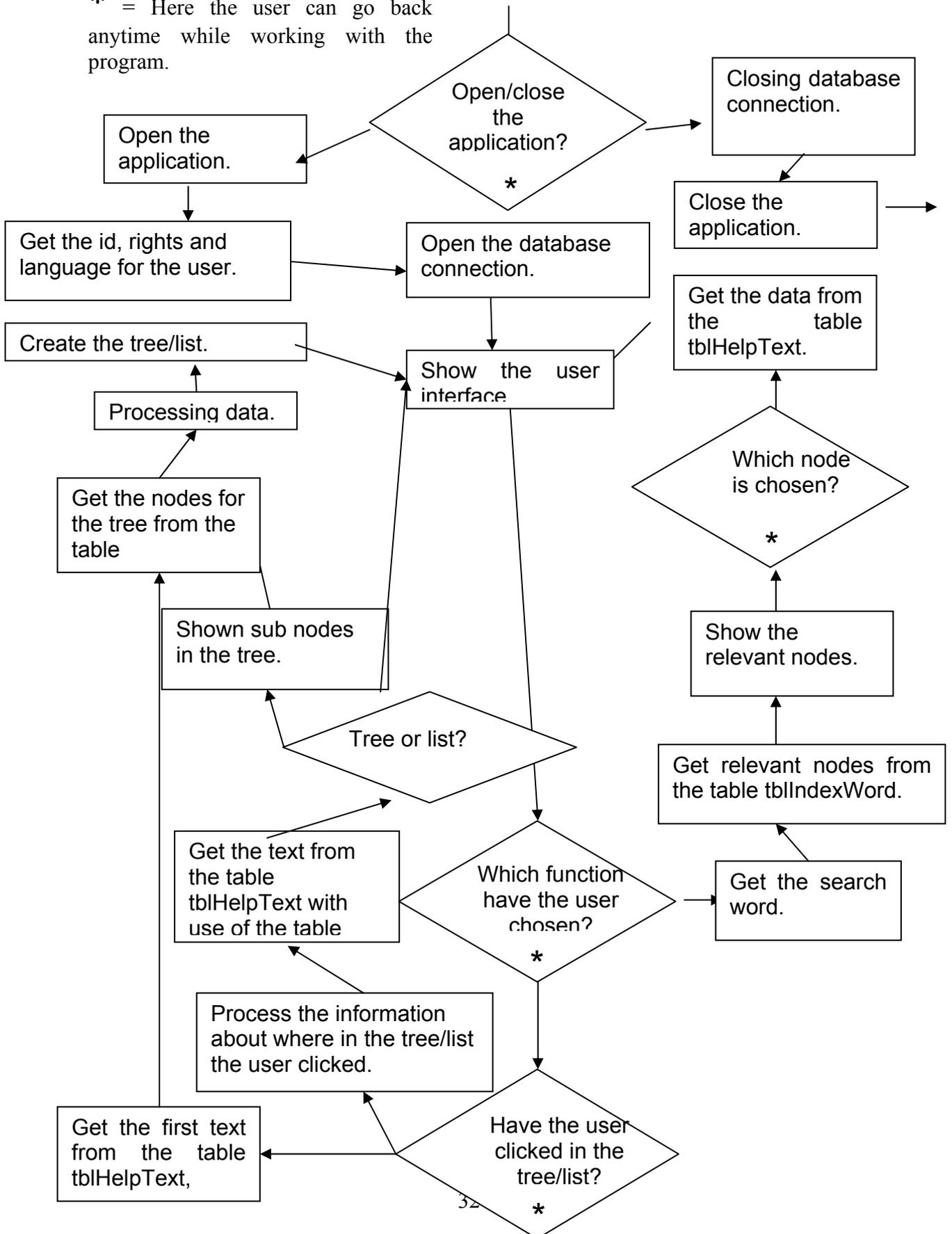
Uppgiftslista och milstolpslista för arbetet med hjälpfunktionen till Virtuella kontoret.

| Uppgifter | | Beroende |
|------------------|--|-----------------|
| T1 | Arbete i andra projekt | |
| T2 | Analys av projektet | |
| T3 | Arbeta fram en projektbeskrivning | |
| T4 | Analys av Virtuella kontoret | |
| T5 | Risikanalys | T2 |
| T6 | Design av hjälpfunktionen | T4 |
| T7 | Skriva koden till hjälpfunktionen | T6 |
| T8 | Sätta samman koden till ett fungerande system | T6 |
| T9 | Skriva uppsatsen | T1 |
| T10 | Skriva loggbok över kodningsprocessen samt dokumentera koden | T6 |

| Milstolpar | Beroende | Datum |
|-------------------|--|-------------------|
| M1 | Preliminär projektbeskrivning | 2003-02-15 |
| M2 | Karta över Virtuella kontoret | T4 2003-03-05 |
| M3 | Överskådlig beskrivning av Virtuella kontorets filer och uppbyggnad. | T4 2003-03-05 |
| M4 | Kravspecifikation | T4 2003-03-17 |
| M5 | Dokument som beskriver risikanalysen | T5 2003-03-17 |
| M6 | Projektbeskrivning | T3 2003-03-19 |
| M7 | Projektplan | T4, M5 2003-03-19 |
| M8 | Dokument som beskriver designen av hjälpfunktionen. | T6 2003-03-28 |
| M9 | Halvtidsinlämning uppsatsen | 2003-04-21 |
| M10 | Grunden till all kod klar | T7 2003-04-30 |
| M11 | Uppsatsen klar | T9 2003-05-20 |
| M12 | Redovisa för Empir AB | T8 |
| M13 | Redovisa för HTU | M11 2003-05-26 |

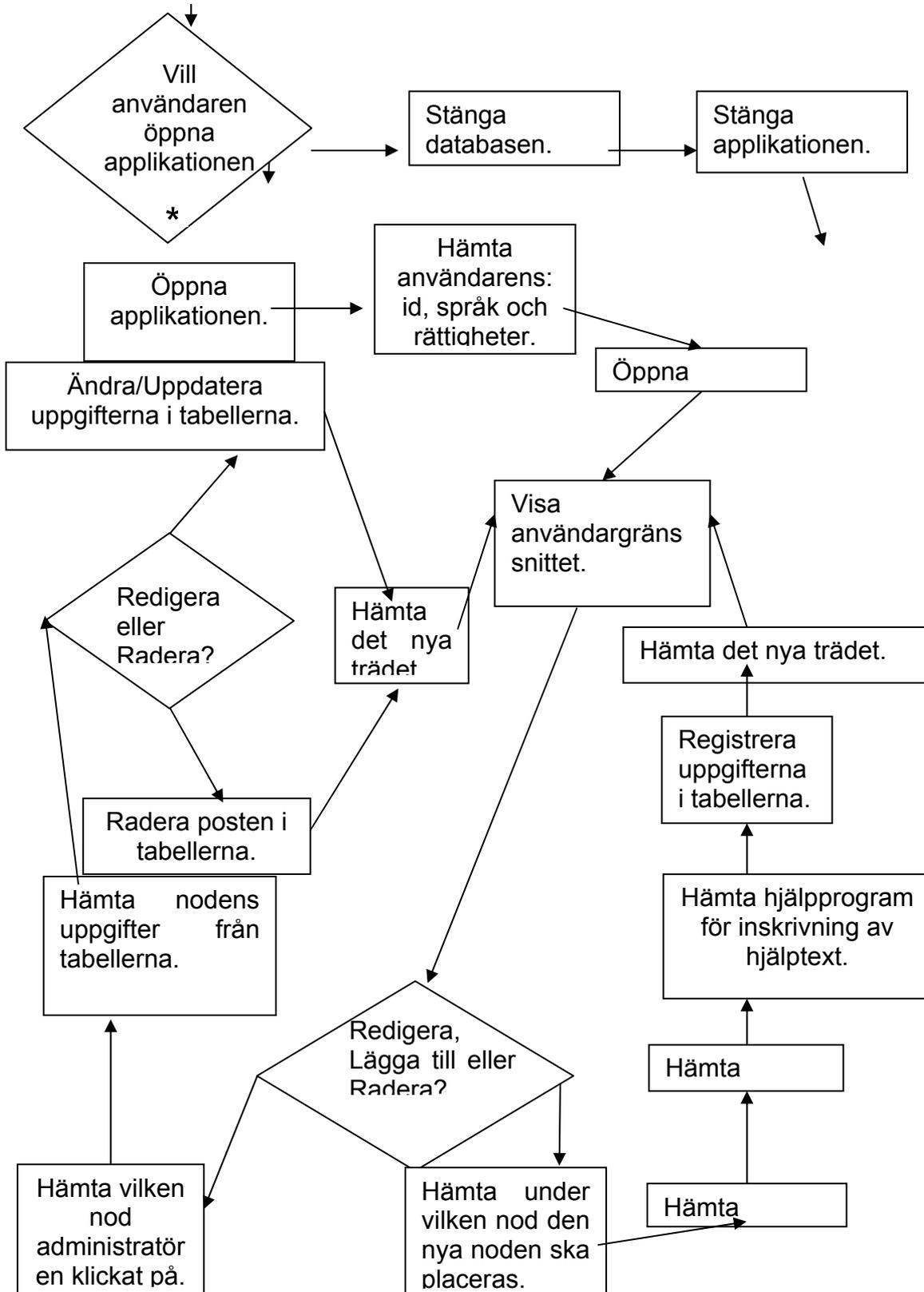
Appendix D –Program flow

* = Here the user can go back anytime while working with the program.



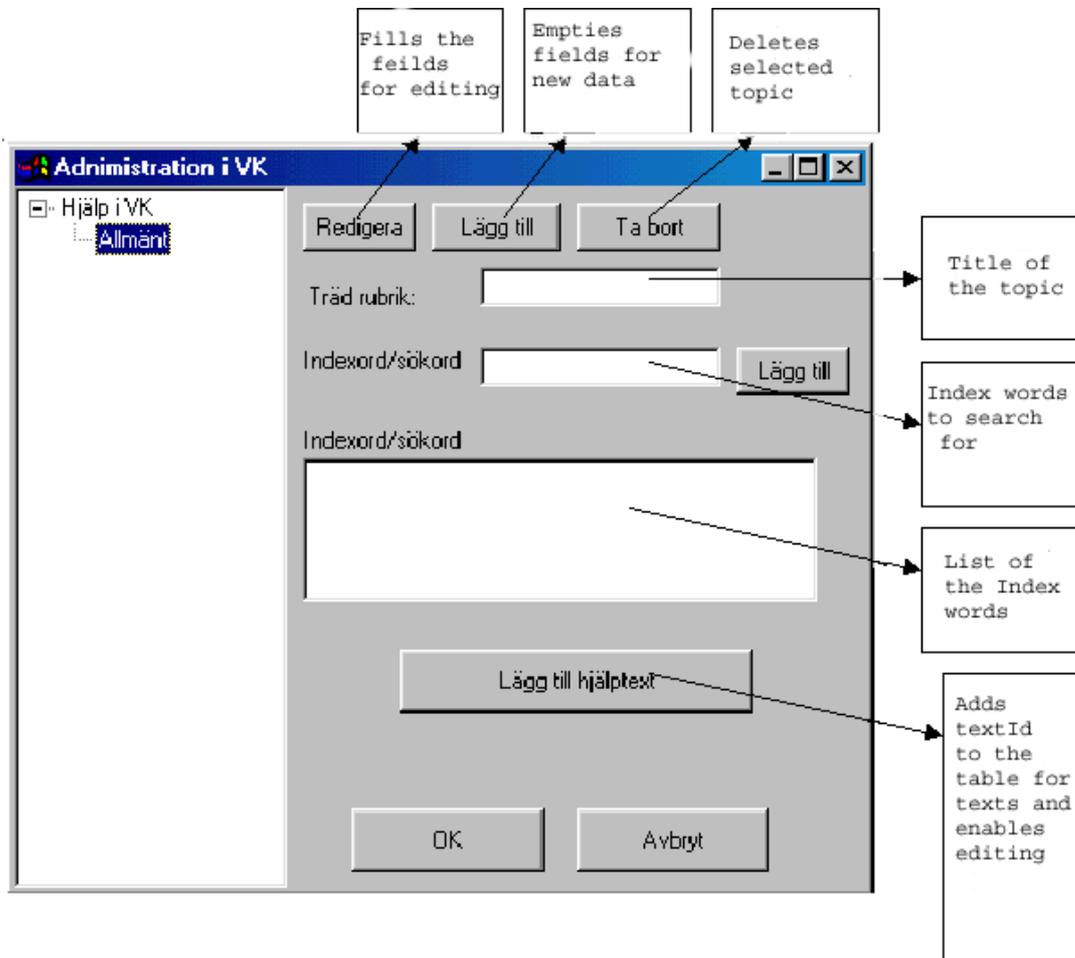
Programflöde

* = hit kan användaren gå tillbaka när som helst under programmet.



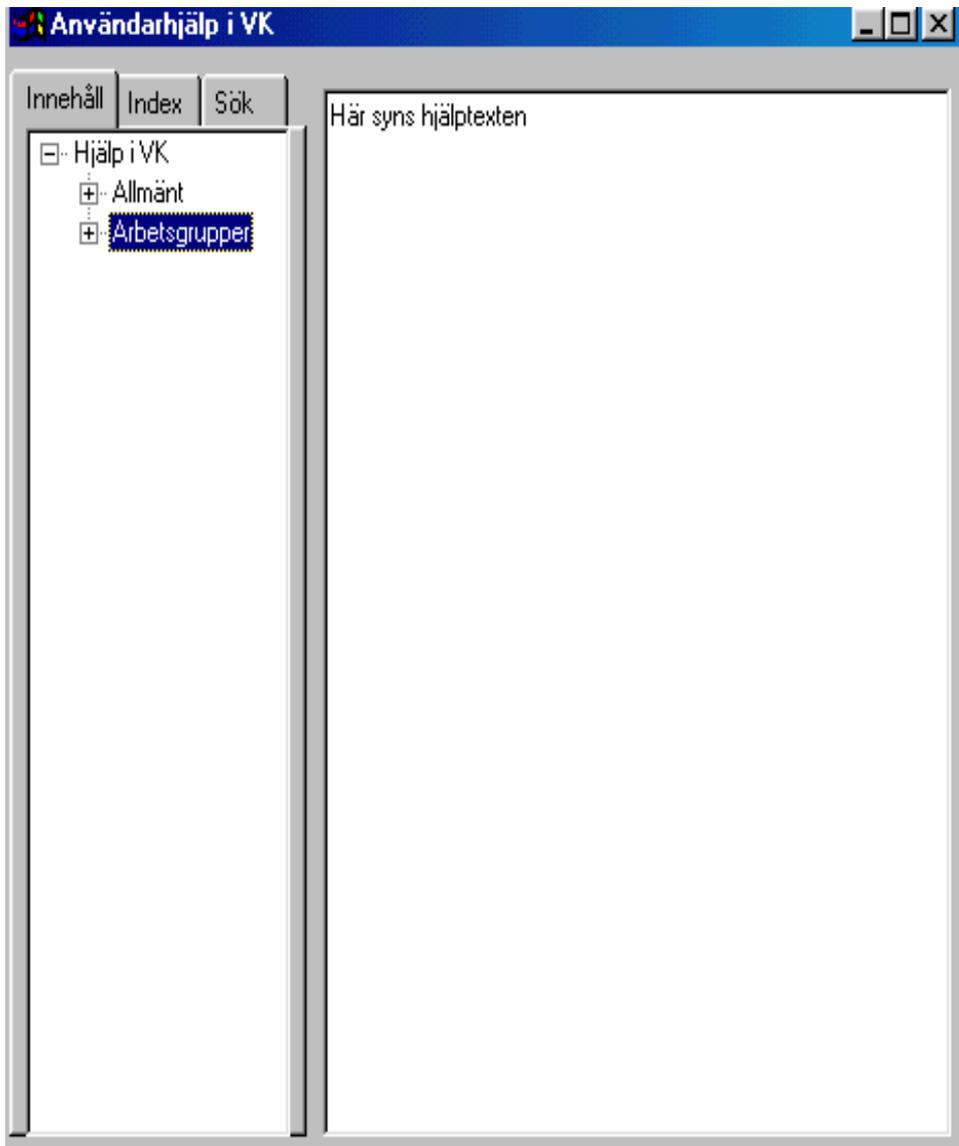
Appendix E - Prototypes

The prototype of the administrator interface:



This is an example of how to design the administrator tool.

The prototype of user interface



Appendix F Additional questions to Survey

Undersökning Virtuella Kontoret

Vilken/Vilka är den vanligaste funktionen/funktionerna du använder? Nämn max 3 saker.

Vad har du mest problem med när du använder VK? Nämn max 3 saker.

Vet du hur du använder alla funktioner och vad de är till för?

Ja ___ Nej ___

Om nej: Vilka funktioner vet du ej hur man hanterar?

Hur skulle du lättast få reda på hur alla funktioner fungerar?

Vad saknar du för funktion?
