Abstract

The master's project described in this thesis includes designing an interface for a Web site and doing usability tests at three different occasions. The designing process was based in the findings from a prior study [Backlund, 2000], design guidelines and checklists and the experience of the interface designer, who also is the author of this master's thesis. Three different methods were used for the usability tests. The Web site that was created is a job site where the users can search for job postings, post their CVs etc.

When designing the interface, the designer tried to follow general guidelines. There are different types of guidelines, some are detailed and can be applied without thinking and the most general, for example Shneiderman's eight golden rules [Shneiderman, 1998], apply to all situations but require an analysis of the actual situation. There were three sets of usability test during the process and in between those, some small usability test were made when the designer felt insecure about which solution to choose. In this study the interface designer also did all programming too, which distracted him some from concentrating on the interface.

When doing a usability test the test user should be matching the target group profile as closely as possible, in this case be interested in his or her own career. If the observer is the one who created the interface it is best if the test user does not know this and the observer must be able to take negative critique. Persons with a close relation to the observer should not be chosen as test users since it might have the affect that the test user is not honest in his or her feedback. The one thing that was found to give best feedback is for test users to talk out loud. If the test user can do this and feel comfortable doing it, it is an excellent method for usability testing.

In this study three methods were used in the usability tests. The methods were codiscovery learning, where two users cooperate and learn together while doing tasks, retrospective testing, where the session is video recorded and the user and observer later watch the session together, and finally *"normal"* usability testing with an observer. In this study the codiscovery learning was not very effective since the two users did not know each other since before. The test user who did the retrospective test was so good at talking out loud that there was no need to watch the video recording of the test, which is the point with such tests. In all types of test the test users were given three tasks to do and after that they filled out a questionnaire and were interviewed.

While doing the test there were some major usability problems found. The Web site had an options page that was very difficult to use, the sign-up procedure did not say clearly that an account would be created and the site was too international, that is, trying to please users from different countries. There were also about 19 details in the interface that need to be improved. The users were in general pleased with the navigation, they thought it was easy to find things and appreciated the speed of the Web site. The interface is available on http://id-2714.nada.kth.se.

Sammanfattning

Webbgränssnitt och användbarhet

Detta examensarbete handlar om skapandet av ett gränssnitt till en Webbplats och användbarhetstester i tre omgångar på detta gränssnitt. Utformningen av gränssnittet baserades på en förstudie [Backlund, 2000], riktlinjer och checklistor för grässnittsdesign och erfarenhet hos gränssnittsdesignern som även är författare till denna rapport. Tre olika metoder användes för användbarhetstesterna. Webbplatsen som skapades är en s.k. jobbsite där användarna kan leta efter jobbannonser, lägga in sina CV etc.

När gränssnittet skapades försökte gränssnittsdesignern följa allmänna riktlinjer. Det finns olika sorters riktlinjer, en del är detaljerade och kan appliceras rakt av och de mest generella, t.ex. Shneidermans åtta gyllene regler [Shneiderman, 1998] kan användas i alla situationer men kräver en analys av den aktuella situationen. Tre omgångar användbarhetstester gjordes under designprocessen och mellan dessa omgångar gjordes mindre tester när designern behövde återkoppling från användare om vilken lösning som borde väljas. I detta examensarbete gjorde gränssnitts-designern även all programmering, vilket distraherade honom något från att koncentrera sig på gränssnittet.

När man gör en användartest bör testanvändarna passa så bra som möjligt in på den slutliga målgruppen, i detta fall bör de vara intresserade av sina karriärer. Om observatören även är den som skapat gränssnittet är det bäst om testanvändaren inte vet om detta och observatören måste klara av att höra negativ kritik om sitt gränssnitt. Personer som känner observatören väl bör inte väljas till testerna eftersom det kan hända att de blir mindre ärliga i sin kritik. Det som gav bäst återkoppling var om testanvändaren kunde *”tänka högt”*, dvs. säga sina tankar när han eller hon gör testet. Om användaren kan göra det utan att det känns onaturligt, är detta en utmärkt metod för användbarhetstester.

I detta examensarbete användes tre metoder för användbarhetstesterna. Metoderna var *codiscovery learning*, som innebär att två användare samarbetar och lär sig tillsammans när de utför vissa uppgifter, *retrospektiv testing*, där test-sessionen videofilmas för att sedan gås igenom av observatören och användaren, och normala användbarhetstester, dvs. en enkel observation av en testanvändare som använder gränssnittet. I detta examensarbete blev testet där codiscovery learning användes inte så effektivt eftersom de två användarna inte kände varandra sedan tidigare. Testanvändaren som gjorde testet där retrospektiv testing användes var så bra på att tänka högt att det aldrig blev aktuellt att använda video inspelningen för att förtydliga eventuella oklarheter, vilket är meningen med denna testmetod. I alla typer av test fick testanvändarna tre olika uppgifter att genomföra och när de var klara fyllde de i en enkät och blev de intervjuade.

Under testerna hittades tre allvarliga användbarhetsproblem. Webbplatsen hade en sida för inställningar som var för svår att använda, i registreringsproceduren angavs inte tydligt att ett nytt användarkonto skapas och webbplatsen var för internationell, den var utformad så den skulle kunna användas av användare från alla länder. Ytterligare 19 användbarhetsproblem hittades i gränssnittet, men dessa var mindre allvarliga detaljer. Användarna var i allmänhet nöjda med navigeringen och hade lätt för att hitta saker och uppskattade att webbplatsen var snabb. Gränssnittet kan ses på http://id-2714.nada.kth.se.

Contents

1 Introduction 9
2 Literature Study 11
2.1 Information Architecture 11
2.2 How to Plan a Usability Test 12
2.3 Methods for Usability Testing 13
2.4 How to Perform a Usability Test 14
2.5 Analyzing Results from Usability Tests 16
2.6 Guidelines and Checklists 16
3 Methodology 23
3.1 The Design Process 23
3.2 Usability Testing 29
4 Results 33
4.1 Comments from Users 33
4.2 Questionnaire Results 35
4.3 Usability Issues and Suggested Improvements 35
4.4 Redesign and Verification of Solutions 39
5 Discussion 43
5.1 The Methods 44
5.2 More Functions for the Web Site 44
5.3 Future Improvements 45
5.4 Comparison of Results from the First Study and this Study 45
6 References 47

Appendix A

1 Introduction

This report is about a study in human-computer interaction and usability as a master's project. The student Jens Backlund, who has the role of interface designer as well as observer and interviewer in the usability tests that were conducted, wrote this report. In this report the author refers to himself when mentioning the interface designer, the observer, the interviewer etc. Supervisor is Ann Lantz, Ph. D. in psychology. The master's project is done at CID, Center for User-Oriented IT Design. CID is part of the Department of Numerical Analysis and Computer Science at the Royal Institute of Technology, Stockholm, Sweden.

The issue is to design a prototype of a job site. The goal is to make an interface as useful as possible for this Web site. The Web site's target group is job seekers in certain professions that can search for jobs, post their CV etc on this Web site. It is a fairly complicated interaction and amongst the thousands of job sites that exist today most have more or less poor usability, but there are a few that are very good. The result from this master's project is meant to turn into a start-up company in the future and one of its strategic advantages will be its usability.

Most Web sites for job seeking have more or less usability problems. This depends on underestimation of usability issues of course, but also because job seeking as a process is complicated. Recruiters need lots of information to find appropriate candidates and therefore many job sites are swamped with input fields that have to be filled out by the job seeker. Many of them make no sense to the job seeker and the process takes long time. Many job sites seem to focus much on the recruiters and less on job seekers. There are actually two target groups, both job seekers and recruiters. As this report is about the part of the Web site for job seekers, the studies focus on them but recruiters are also included.

To reach the goal of creating a useful interface you must follow the design guidelines and involve users in the design process. Their feedback will help understanding usability issues for the interface so it can be improved during an iterative process. The feedback will also determine if the interface in fact is useful. To determine this, you must consider subjective feedback from questionnaires, users' comments and how many observed difficulties or errors are found while doing usability tests. The usability can be measured in time for certain tasks, number of errors, etc. Varying usability test methods will be used to measure and understand the usability of the Web site.

Before this master's project had started a study [Backlund, 2000], much like this, was conducted on the same issue. Jens Backlund then at University of Chile, Santiago, Chile, wrote that study as well. That study is referred to as the *first study* in this text and the results from the first study and *this study* are compared. Supervisor at University of Chile was Jaime Sánchez, Ph. D. in informatics and education. Sánchez has designed the questionnaire used in this study [Sánchez, 1999]. The first study, conducted at University of Chile, was the author's first course in human-computer interaction so this master's project is on a higher level. A market research was also done as a part of an Internet marketing course at University of Chile with Máximo Bosch, Ph. D. in operational investigation, as supervisor

Usability is what laymen call user friendliness. Though it is not about friendliness, it is about products and systems being usable and easy to use.

If a system can be used to achieve a desired goal it is useful. Usefulness consists of utility and usability, see *Figure 1.1*. Utility is that the system's functionality can do *what* is needed. Usability is *how well* the users can do that functionality.

Figure 1.1 *The definition of usability by Nielsen [1993].*

One can measure usability, using usability criteria, for example how fast a user can do a task or how many errors a user commits. This paper focuses on understanding usability and interfaces rather than measuring them in numbers.

A related issue is accessibility, whether the system is accessible for users with disabilities, for example blind or colorblind people. On the Internet accessibility also concerns users with different browsers. If a Web site uses images to give information or for navigation, people using text-only browsers will be shut off.

2 Literature Study

The literature study deals with what has been written about usability and user interface design. The focus is on information architecture, usability testing, for example how to plan a test and descriptions of some of the methods that exist, and rules and guidelines for user interfaces. Different rules and detailed guidelines are compared and analyzed.

2.1 Information Architecture

Information architecture is about organizing information. you can organize information in many different ways, alphabetically, by size, by date etc. Users who look for information will typically try to find the information they need using different methods. Some like to search for specific keywords and some prefer browsing to search. An example of a directory is Yahoo! where information is organized in categories and subcategories.

You should consider the users' expectations as they interact with your Web site. If you have a large amount of information to organize it might be a good idea to get some help from the users. One method is to ask some users to organize the contents of your site in a way that makes sense to them. Write down all content items on one card each then ask users to organize those cards in piles of related content items and assign a label to each pile. You should do this with several users and colleagues in your own organization. Then you compare and analyze the results and base your site's structure on your analysis.

When organizing a Web site, there is a need for balance between breadth and depth of the site. A broad site typically has many options on the first page, and there will exist a risk that the user is overwhelmed, but the user can get to the information he is looking for in a few clicks. A deep site requires many clicks to get to the content you are looking for. An example of a deep site is Yahoo! Some Web sites may work best deep and some broad but a balance is needed. Web sites with a huge amount of information might of course end up being both deep and broad. Big Web sites can be divided into sub sites, for example one for each target group. They can also use shortcuts.

Shortcuts are needed as a complement to the navigation if a site is deep and it takes the user many clicks to get from one location to another. According to Jennifer Fleming [Fleming, 1998] there are several types of shortcuts:

- **Searching.** There should be a search box on *every* page of the site.
- **Table of contents.** You should include a clickable list of categories and subcategories. It makes the structure of the site clear to the users and makes it easy to jump to another location.
- **Index.** An index is an alphabetical list of all contents. Some users find indexes easier to use than search boxes and tables of contents. One problem is that users may look for a word that is not in the index because the authors thought of a synonym to the word. For example *site structure* instead of *site organization*. Synonyms must be included to make the index useful.
- **Site map.** Site maps are like table of contents but graphically presented. They are good for users who are visually oriented but may be slow if the graphics are heavy.

- **Pull-down menus.** A pull-down menu can include locations, for example. It expands when clicked on. They save screen space but all users may not see them. According to Jakob Nielsen pull-down menus often cause more trouble than they are worth and can be confusing because they are used for several different interaction methods. Scrolling menus reduce usability when users cannot see all their options in a single glance.

One has to remember that shortcuts are a supplement and are not sufficient alone [Fleming, 1998].

2.2 How to Plan a Usability Test

First of all, you need to know why the test is being done. Is it to measure the overall quality or to learn about details in the interface? If you want to estimate the overall quality a *Summative Test* is recommendable, it means that you measure the number of errors the users commit or the time it takes them to do certain tasks. The other approach is a *Normative Test*, which is better if you want to improve the user interface and to learn about usability details. A typical way to conduct a normative test is to observe users and ask them to talk out loud about what they are doing [Nielsen, 1993].

Jakob Nielsen recommends that you do a *Test Plan*. These are the things you need to consider in the plan, according to Nielsen:

- The goal of the test, what do you want to achieve?
- Where and when will the test take place?
- How long time it is estimated to take?
- What computer support is needed?
- What software is needed?
- What should be the state of the system at the beginning of the test?
- What should system load response times be? Should the system be slowed down to make it more realistic?
- Who will lead the tests?
- Who will the test users be? How will you get hold of them?
- How many test users are needed?
- What tasks will the users perform?
- What criterion is used to determine when a task is done?
- What user aids will be available to test users?
- How much help can the test leader give to users?
- What data will be collected? How will the data be analyzed?
- How do you determine if the interface is sufficiently good?

The average number of usability problems for projects in general is 41 and the chance to discover each one of them is 31 % for one single user. Jakob Nielsen discovered this in an investigation he did. He recommends five users for a usability test and that you do many tests as the design process keeps going. Others say four to five test users is optimal [Webmonkey].

Before starting with the test you should do one or two so-called *pilot tests.* Since users can be difficult to get hold of and this test is more a test of the testing procedure than the interface you could chose a colleague to do the pilot test. This preparing test will help in discovering how long time the test will take, if the instructions and questionnaire are understandable and if the tasks are too easy or too difficult.

2.3 Methods for Usability Testing

Nielsen describes many methods for usability testing and some of them, which can be suitable for this project, are described here. In previous tests in the first study observations, interviews and questionnaires were used.

2.3.1 Observation and Talking Out Loud

In an observation test the user does a few tasks or uses the software or Web site as usual and the observer takes notes of what he does. It is an excellent method to encourage users to talk out loud what they are doing and thinking. When verbalizing their thoughts you can discover how they perceive the system and perhaps discover any misconceptions. Not everyone feels comfortable doing this of course, which is one of the disadvantages with this kind of test. If the user becomes silent the observer can ask the user what he or she believes something means.

You should try to observe the user in a normal situation. Therefore you should not give help to the user since he or she normally does not have an expert in the room to ask. At the end of the test you can start giving help, to encourage questions. It is good if the test can be conducted in the users' own environment, which makes it easier to reveal the users' tasks. The observer should be as quiet as possible. you can also do a usability test of competitors' products or Web sites. What the user does is more important than what the user says. What the user says *while* doing the test is more useful than what the user says *after* the test.

Another advantage is that this kind of test is cheap and gives high quality results. The disadvantages are that it might be difficult to make appointments with users and that most users feel uncomfortable talking out loud. This test method is not appropriate for measuring performance.

2.3.2 Codiscovery Learning

Codiscovery learning is like the previous method but there are two users. They cooperate while doing tasks. The advantage of this method is that they talk and perhaps explain things to each other. This way you will get excellent feedback of what the users think, how they would explain the interface etc. It is more natural to explain to someone than to talk aloud, which is the main advantage. The disadvantages are that it requires twice as many users and that some users just cannot cooperate. It is a good for testing interfaces for children since it might be difficult to get them to talk out loud.

2.3.3 Retrospective Testing

In retrospective testing the testing session is being recorded and afterwards the observer and the test user go over the video together. The observer can stop the tape and ask for a detailed explanation without interfering with the test. You can extract more information per user this way but it takes more time. If there are few users this could be a good idea. The users might give better comments while watching the videotape than they gave during the test because the user is not occupied with the tasks anymore.

2.3.4 Focus Groups

Focus groups are an unstructured way to assess users' needs and feelings. They are lead by a moderator who keeps the discussion flowing. There are six to nine users in the group. Focus groups can be used in the beginning of a project or after the product is finished and has been used for a while. Focus groups give spontaneous reactions and the group dynamic might help get extra information. When someone has to defend a viewpoint more emotional reasons can be discovered. The disadvantages are that they are difficult to analyze and have low validity. At least

two focus groups should be held to get more accurate results. Focus groups are quite demanding in terms of number of persons needed.

When preparing a focus group the moderator needs to set goals and make a list of issues that are to be discussed. To make a detailed analysis is difficult but the moderator can write a simple summary.

2.3.5 Interviews

Interviews are well suited for exploratory studies when you do not know what you are looking for. Subjectively pleasing is a part of the usability definition. It is possible but very difficult to measure objectively, so are opinions and anxieties. Therefore asking the user is the best alternative. It is an excellent method for knowing what the users want. The questions should be open and not just "*Did you like it, yes or no?*". You should use the opportunity to rephrase or follow up questions when it is needed. Asking about critical situations that have happened to the user, anxieties or opinions often give good feedback. You can get more in-depth attitudes by interviewing. The disadvantages are the time; it is time consuming to do them. It is also difficult to analyze and compare the results.

2.3.6 Questionnaires

Questionnaires give the subjective satisfaction and opinions of the respondents. If you do a questionnaire it is important to have something to compare the numbers with, for example the results of the same questionnaire done for the same product in an earlier stage, a goal decided in advance or for a competitor's product. You must test the questionnaire first and only ask questions you want the answer to. A questionnaire should never take more than 20 minutes to fill out, which is a well-known fact in marketing where questionnaires are used for market research. When doing testing in the first study [Backlund, 2000] of the development process of the Web site it was found that questionnaires give least information of the methods listed. You should have at least thirty persons answering the questionnaire and some amount of work is needed before doing it. However once the questionnaire is sufficiently good it is easy to have several persons fill it out.

2.3.7 Task analysis

You may be in an early stage of the design or you may want to do a task analysis, that is, to find out what the users wish to accomplish. If that is the case you can do a focus group, which can take the form of a brainstorming with your intended users as participants.

The purpose of a task analysis is to know:

1. Users' overall goals.
2. How users currently approach the task and the underlying functional reasons.
3. Users' information needs.
4. How users deal with exceptional circumstances or emergencies.

Task analysis can also be done by interviewing and observing users in their natural environment. The users' models of a task may give rise to useful metaphors. Users who are especially effective, maybe because they have a trick to perform certain tasks, are interesting as they can give inspiration to new ideas [Nielsen, 1993].

2.4 How to Perform a Usability Test

First of all test users need to be recruited. They should be offered some compensation, especially if they have to leave their offices during working hours. They should also be told about the test, what it is about and how it will be conducted

so that they think the test is interesting instead of being intimidated by it. You ought to encourage them too, make them feel important and feel that their feedback is valuable. There is a big risk that users postpone or cancel tests but if you make the user feel important and the test interesting that risk will be smaller [Dumas & Redish, 1993].

When you start a test the user should receive written instructions and you start when it is clear that the test user has understood and has had a chance to ask questions. The user then does his tasks while being observed. The observer must try to interact with the test user as little as possible, especially if the observer is the designer. In that case there is a risk that the observer/designer will try to excuse the interface or help the user to make the interface appear easy to use. If the designer performs the usability tests he or she must be able to accept negative critique of the interface.

After the tasks have been done or the user has chosen to stop, the user could fill out a questionnaire and/or be interviewed. The questionnaire should be filled out before the interview if both are done, since the interviewer might influence the test user. There is a big risk the interviewer will influence the test user merely by asking certain questions or by reacting to the user's answers, this would of course influence the interview results. If the questionnaire is done first the only influence when the test user fills out the questionnaire is the questionnaire itself, and not a previous interview. During the interviews the interviewer must try to be as neutral as possible.

After the test is finished the observer should try to have an informal discussion with the user about the results of the conclusions and the interview. That way you could get feedback again about the conclusions that were made. It also serves as a form of thank you to the user; it is interesting to see that you made some difference. It was discovered in earlier usability test that this type of informal discussion gives much useful feedback.

What should you be looking for when doing a test? You can note where users pause for guidance, how and when they use search boxes and site maps, when they use the back button in the browser, if they understand menus and labels and whether they leave the site and come back. You should ask why and how the user does certain things [Fleming, 1998].

It is generally recommended that the tests be carried through at the users' locations, their homes or offices, depending on where they would use the service. Of course it is not always possible and sometimes usability labs are available, which can be even more useful in some situations.

Video recording or audio recording can be a useful instrument. It can be effective to let developers and designers watch someone having problems using their sites or software. However recordings can make users uncomfortable so you must inform the user before about recordings and only do it if the user feels comfortable with it [Nielsen, 1993]. There are programs that record a video sequence of the screen but these programs can do more harm than good since they have to be restarted every few minutes because of low memory. An alternative is to record to a normal video recorder from a graphic port on the computer, if that is possible.

It is also possible to evaluate the usability using questionnaires on the Web site, the log files for the Web site and interviews over the phone.

2.4.1 How to Make the User Comfortable

There are a few things you should do to make the user feel relaxed and comfortable with the test. The user must be aware that it is not the user that is being tested, but the interface. Some prefer to say evaluators instead of test users to emphasize this. They should also be told that the interface could be difficult to use so they will not feel stupid if they cannot use it and that they can stop the test whenever they feel like it.

Before the test begins recordings should be explained. The test can begin when all questions from the user are answered. You should start with an easy task and offer the user coffee to make him feel relaxed. More tasks should be handed out one at a time.

The results should always be confidential and names should never be used to refer to users.

2.5 Analyzing Results from Usability Tests

Everyone wants everything. You can never please all users. It is also important to know that users may say one thing and do another [Webmonkey]. If they say they would use an interface in a special way, but *do* use it in another way you should go by the way they *did*, not they way they said. A study conducted by Root and Draper [1983] showed that some users said they did not know certain commands and later commented the same commands in a questionnaire. So they obviously knew at least something. Data about people's actual behavior should have precedence over people's claims of what they think they do [Nielsen, 1993].

If you do a summative test you will get data that is easier to measure and analyze than qualitative results from observations and interviews. Insight is more important than data so early in projects normative tests should be chosen. It is important to look for patterns in responses and not just implement all wishes [Fleming, 1998].

You should use your judgment and be aware that it is easy to discard feedback because you think differently or because it expresses negative critique against your interface, if you are the designer that is.

Weaknesses of usability testing are that the tests are usually too short to test the whole interface and that every test user sees the interface for the first time, at least if it is a new product. You will not know how it is to use the interface after several months unless you make efforts to specifically test that.

2.6 Guidelines and Checklists

There are many guidelines and rules for interface design. There are also checklists that might be easier to use for some designers. This section compares Shneiderman's [1998] eight golden rules of interface design, which is a very general set of guidelines, with more detailed ones from IBM's department *Ease of Use* [Ease of Use] and Nielsen [1993].

2.6.1 The Eight Golden Rules

Each of the eight golden rules are explained and compared to similar rules from the other sources.

1. **Consistency**. Identical terminology should be used in prompts, menus and help screens. Also colors, layout, capitalization, fonts and so on should be consistent. Exceptions should be limited in numbers.

IBM's correspondent rule is called *familiarity*. Build on users' prior knowledge, especially experiences from the real world. If the system is consistent the user can learn an action once and apply in many different situations [Ease of Use]. Metaphors can be used but in many cases they are bad, since designers sometimes take them too far or get restricted by them.

Consistent systems will make the user dare to explore the system and learn more. The same info should always be in the same position and look similar. If the user knows that the same command always has the same effect he will be more confident when using the system [Nielsen, 1993].

2. **Shortcuts**. Enable frequent users to use shortcuts. Response times should be short.

   When choosing how many alternatives there should be for a user interaction, you should make the easiest alternative visible so new users can use that and do not have to choose between alternatives. When users become more experienced they can be offered more alternatives. This way they can choose the most effective way when they are more confident with the system [Nielsen].

   There should be a menu with the commands that were used latest so the user can easily repeat them [Nielsen]. Microsoft Word has an example of this, in the *File* menu there is a list of the files that were opened latest. Clicking any of them opens that file.

3. **Feedback**. Offer informative feedback for every user action. Objects involved should be presented visually.

   The system, software or web site should always keep the user informed of what it is doing and how it interprets the user's input. When the user is about to commit an irreversible action there should be a warning message informing the user about this [Nielsen, 1993].

   Response times should always be as fast as possible. The limit to what is perceived as immediate is 0.1 seconds. The limit for perceiving something as flowing is one second and after about ten seconds the user will lose attention on the current interaction. When something takes more then ten seconds there should be a visual feedback showing how much time is remaining and how much work has been done. For actions that take two to nine seconds there could be some feedback that is not drawing the attention very much, for example a small number increasing in a corner of the screen [Nielsen, 1993].

4. **Design dialog to yield closure**. Sequences of actions should be organized in groups. The informative feedback at the completion of a group gives operators the satisfaction of accomplishment, a sense of relief.

   Create a feeling of progress and achievement. Immediate feedback allows users to assess whether the results were what they expected. Previews of coming results should be available where possible [Ease of Use].

5. **Error prevention and simple error handling**. Design the system so that the user cannot make a serious error. If a user commits an error, the system should detect the error and offer simple, constructive and specific instructions for recovery. The user should not have to redo the entire action, only the erroneous part.

   IBM puts it this way; the design should keep the user out of trouble. The interface should provide visual clues, reminders, list of choices and other help. It can be automatic or on the user's request. The system should remember things so users do not have to.

   Nielsen has some detailed advice about error handling. You should try to avoid situations where the user can spell wrong. For example if there are only a few options you should have a select box instead of having a text field.

   Usability tests and error logs give information about how frequent and how severe certain errors are. Using this information you can improve the interface.

   It is very difficult to prevent errors totally, especially if the system is advanced. If the user does commit an error there are four rules for good error messages. It should be in clear language. The common example "*Error 404*" is not clear language and does not give the user clues to what went wrong. The message should contain constructive advice on how to solve the problem. The message should be precise, not vague. The last rule, as important as the others, is that the message should be polite and not blame the user nor intimidate the user. The system should take responsibility for the error. A good error message could be "*We are sorry, but we need more information to complete your request, please fill out your e-mail address in the field below and press submit*."

   One can use levels in the error message. Some Windows programs use this, first there is a simple error message and if you press a button labeled "*error details*" you get more details about the error. If error numbers are needed to give to support personnel it should be given in the details [Nielsen, 1993].

6. **Permit easy reversal of actions**. This feature relieves anxiety, since the user knows that the error can be undone.

   IBM recommends that you encourage users. Make actions predictable and reversible. One user action should have one effect, and that effect should of course be predictable. This has also to do with rule one, consistency.

   Users should feel confident in exploring, knowing they can try an action, view the result, and undo the action if the result is unacceptable [Nielsen, 1993].

   On the Web, people often get frustrated by not being able to go back or by feeling trapped. Older Web browsers have a usability problem, if a user fills out a form, submits it and then clicks back, maybe because the user discovered an error, the form on the first page will be empty and the user will have to fill it out all over again. In newer browsers the contents of forms are saved so that when you clicks back the contents are still there.

7. **Control**. Users want to feel that they are in charge of the system and that the system responds to their actions. Avoid surprising system actions.

   Place the user in control and provide proactive assistance. Enable users to accomplish tasks using any sequence of steps they would naturally use. Current state and possible actions should be obvious. You can leave the system and return later and the system is in the same state [Ease of Use].

   IBM also talks about personalization. Allow users to personalize. No two users are alike. Customization can help make an interface feel comfortable and familiar. Personalization gives higher productivity since user can change for example default values and thus do not have to change every time they do an action. In an environment where multiple users share an advice, allow users to create their own system personality and make it easy to reset the system. In an environment where one user may be using many computers, make personalization information portable so the user can carry that "*personality*" from one system to another.

8. **Reduce Short-term memory load**. The limitations of human information processing in short-term memory are five to nine things to remember. Displays should be kept simple and multiple displays used. There should be online access to syntaxes, codes etc.

   The user should never have to retype data or have to remember cryptic codes etc. When an error is committed the user should only have to modify it, not retype it from the beginning.

   The screen should contain the information the user needs and nothing else. Every additional element is one more thing to remember, as the golden rule number eight points out |Nielsen, 1993]. On the Web every additional element is also one more thing to download, increasing response time.

Some of IBM's rules are additions that are not included in Shneiderman's golden rules. IBM speaks of simplicity, obviousness, availability and affinity.

Simplicity
Do not compromise usability for function. Functions should be included only if task analysis show they are needed. The utility of a system would be low if it includes unnecessary functions. Actions should be organized in sequences that match the way users want to do something.

Obviousness
Make objects and their controls visible and intuitive. It has to be noted that intuition is based on a web of prior knowledge and not all users and designers have the same knowledge. Instincts on the other hand are something you are born with, but instincts are difficult to design for.

Availability
Make all objects available at all times. Modes should be avoided. For example a system with an "*Edit mode*" where you can only edit and a "*View mode*" where you cannot edit should be avoided.

Affinity
Bring objects to life through good visual design. Visual hierarchies where the most important object is most visible should be created. It is also important to consider affordance, which means that you can see *what things are for*. Use white space to provide visual breathing room.

Things seem to belong together as a group or unit if they are close together, enclosed by lines or boxes, move or change together or look alike with respect to colors, shape and size. When designing the layout of a screen you should try to group things and use colors, shapes and sizes to prioritize the attention of the user to the most important elements first [Nielsen, 1993].

Nielsen has some detailed additions that the others do not have.

Speak the Users' Language
Words and language that users use themselves should be used in interfaces. This goes for icons and graphics as well, since symbols can vary between different cultures. There are statistics over how many percent of populations that know certain meanings of words. You can also ask the users which words they would use, but since many words have too many synonyms it is difficult to find one word that the majority understands. The best method when choosing words is to have developers, usability experts and users suggest words and then let users vote on which word they prefer.

Online Help and Documentation
Any system should be so easy there is no need for online help or documentation, but this is of course not the case in the real world. Even for those systems that are that easy there should be an online help if the user might want to learn more to get to an expert user level. Online help can itself be a usability problem if it is designed badly and is not tested. Users neither read Web sites [Ten Good Deeds in Web Design] nor manuals very carefully, according to a study by Retting to which Nielsen refers [Retting, 1991]. Therefore it is important to make an online help for a Web site small so there will be a greater chance that user will read them.

There are three stages when a user looks for help.

1. Search for information that is relevant to a specific need.
2. Understand the information.
3. Applying the information to the procedure described.

When a user looks in the help it is probably because of an *urgent* need to get help. Therefore it is best with an online help that is task oriented and easy to scan.

Instructions for a certain task should be divided in sequences with a number on each step. There should be examples that the user can modify so they apply to their current situation. Examples are easier to understand than abstract explanations. The quality of the text is one of the most important factors in help systems and technical writers should write the help texts.

The help system should include a search function and an index with terminology from the system, users, competitors and possible synonyms. It can also include table of contents, frequently asked questions, tutorials or tours for beginners.

If the help is being kept in a separate window at the side of the main application window, it is not necessary to remember all steps while doing them.

2.6.2 Top Ten Dos and Don’ts in Web Design

Nielsen has made detailed checklists for Web design. Many designers have problems following guidelines and think it is easier with examples and checklists. This is a list over top ten things to do to improve usability for Web sites [Nielsen, 1999b].

1. Place the logotype in the upper left corner, linked to the homepage.
2. Provide search if the site has more than 100 pages. This is one form of shortcut, see the section about information architecture and golden rule number two.
3. Use simple page titles and headings.
4. Facilitate scanning. People do not read Web pages; they scan them, unless they are very interested in the text.
5. Structure the content in several pages instead of putting it all on one big page.
6. Use product photos.
7. Use image reduction as much as possible.
8. Use good link titles so people know where they will end up.
9. Make all pages accessible for users with disabilities.
10. Do the same as everyone else. That is what users will expect. This recommendation has to do with consistency, golden rule one, for the whole Web.

The ten most common mistakes in Web design today are [Nielsen, 1999a]:

1. Breaking or slowing down the back button. This is breaking the golden rule number seven about giving control to the user.
2. Opening new windows.
3. Non-standard use of GUI elements. This is an inconsistency and violation of the golden rule number one.
4. Lack of biographies.
5. Lack of archives.
6. Moving pages to new URLs.
7. Headlines that make no sense out of context.
8. Jumping at the latest buzzword. You has to consider the utility.
9. Slow server response times.
10. Anything that look like advertising.

As these two latter lists are more detailed and focus on Web usability they are also easier to check than more general guidelines like Shneiderman’s golden rules.

3 Methodology

This section describes the design process, that is, how the interface was created. The creation of the interface was based both on guidelines and user testing methods. Since there cannot be guidelines for every application and since you cannot ask user to design an interface they like, the design process has to be based on creativity and experience as well. The methods used for usability tests are described in section *3.2 Usability Testing*.

3.1 The Design Process

When the design process started one set of usability tests has been conducted which is the base for the design process. Competitive Web sites with inspiring solutions that seem to work well also influence the design.

Hotmail [Hotmail.com] is a good example. It has millions of users and therefore it is reasonable that Hotmail is easy to use and easy to sign up for. On their sign up page there is a form for giving personal information, like the one needed on the pages where the user signs up to post his or her CV.

On Hotmail's [Hotmail.com] sign up page there are yellow help boxes beside every field that is to be filled out, just like the design suggested in the first study, see *Figure 3.1*. Some fields, that are considered difficult and that require more help, have links in their help boxes down to a help text on the same page. Since the link goes to the same page, the user never actually leaves the page. This means that input given in the form is never lost, which can happen if the user goes to another page and comes back.

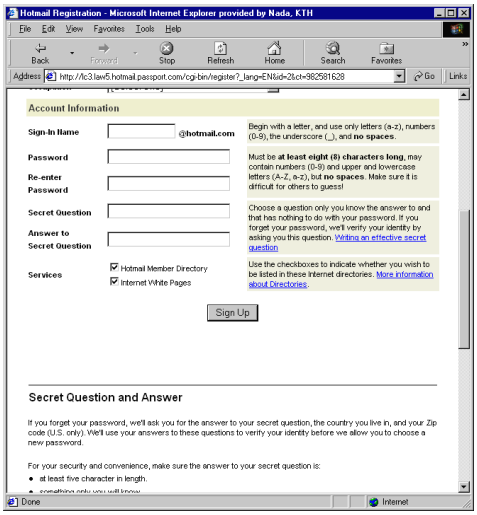

Figure 3.1 *Hotmail's sign up page. There are links in the colored boxes to the right of the input fields that leads to the text shown further down on the same page.*

The interface problems encountered when designing are listed below and the reasoning when solving them are given. When solving interface problems you have to consider guidelines and rules but this is not always sufficient. As an interface designer you also need creativity and experience to come up with a number of possible interface solutions to consider. Another source for the creation of interfaces, besides guidelines, rules, creativity and experience, is discussing with colleagues.

3.1.1 Help System Dilemma

The help system needs to be consistent. It is expected to be used rarely except for the pages where you fill out your CV. All input fields on the entire site have a *Help Box* to their right, see *Figure 3.2*. In this help box there are instructions for filling out the field. If the user fills out a form and then clicks on to a help page, the form might be empty when he returns, if the link used the same browser window.

Figure 3.2 *The search jobs page.*

The user may have spent several minutes filling out the form and of course would not be happy to see that all their work was lost.

This problem is mostly for older browsers. There are three different possible solutions to the dilemma:

1. **Help pages open in new browser windows.** The advantage with new browser windows is that the user can have the help window besides the main browser window when doing some procedure. Forms will not be cleared when going to a help page. Nielsen mentions three disadvantages with opening new windows.

    1. Links are inconsistent, some use the same browser window and some do not. The user is used to links using the same browser window.

    2. The user may not notice that a new window is opened, especially if the original window was maximized to cover the entire screen, in that case the new window will cover the entire screen and therefore the original window will be "*hidden*" behind the new window.

    3. The back button stops working.

    However using a smart technique minimizes these problems. Setting the window size makes the window smaller and it can never be maximized and cover the main window completely. Stating clearly that the link opens in a new window minimizes the inconsistency problem. The most severe usability problem is that the user may not find the help window. If the user clicks the link again it will appear again, however. The back button problem is not a problem *if* the user sees the new window. Hotmail

[Hotmail.com] uses this and they even move the main and help windows to be side-by-side, covering the entire screen.

2. **Normal linking – Help pages in the same window**. As mentioned above the problem is that forms may be reset if the user goes to a help page in the same browser window and returns. You could make a solution that saves the form values when clicking a link and reprints them when going back.

3. **Help at the bottom of a page.** Help at the bottom of the page can be used at the pages with forms, but it should not be used for others that will need less guidance. This solution is really interesting since help will be needed mostly for filling out forms. One hypothesis is that novice job seekers may want to see examples of how to write their CVs.

   On some pages, like the *Post CV* page it might be too much with all the forms, help for the forms, help to format the CV and CV examples.

For the help system the first option was chosen, mainly because it is expected that help will be needed mostly on the *Post CV* page. This page turned out to be very long so help at the bottom of the page would make the page extremely long. Most software have separate help windows.

3.1.2 How to Save the Users' Options

The user may save options, for example default values for the search form, a list of countries and possibly more in the future. There are two ways to save the users' options. One technique is so-called cookies that save information on the users' computer. Internet Explorer and Netscape use different cookies so if the user switches to a new browser or switches computers the defaults will not appear.

The other technique is to save it on the user's account. The problem is then that the user has to log in to make the saved default values work.

One could also let the user choose between the two methods. In that case you would have to explain them and there might be problems if user saves different values on the own computer and on the account.

A hypothesis is that users will stay on the same computer mostly so the cookie technique will be used. To minimize the problem when a user changes computer, users will be made aware of the fact that the information is saved on their computer. The checkbox to be clicked will have the label "*Save these options as default on this computer & browser"*.

3.1.3 Formatting the CVs attractively

The users should be able to format their CVs attractively. It was shown in a market research that job seekers want the CV to look good when they post it on job boards. For users who do not have a CV already it might be best to use predefined forms, that way they will know what to write and it would be formatted automatically.

Formatting is a typical thing to do for experienced users. A novice on Internet or job seeking may just want to post a CV and might be concerned how to do that, without having to worry about formatting it attractively.

Technically it is difficult to make formatting easy in a Web browser and it will not work, at least not the same way, in all browsers. You could solve the problem by letting the users use HTML, which has become a popular technique by being easy to use. All will not be able to format their CVs with HTML, but it can be a feature for

advanced users. In a future version the user may also have the choice from different templates to use for their CVs.

3.1.4 Navigation

At first the site structure was several pages in a tree structure. There was a problem about how to build this structure. Test users were asked to sort the contents of the site in piles and this showed that it could be done a lot simpler by just making three sections of the site.

The sitemap, shown in *Figure 3.3*, is accessible from all pages. A site search box is not included since it would perhaps be confused with the job search box.

Figure 3.3 *The structure of the Web site.*

The first page shows the most important contents of the Web site; job searching, log in and sign up for the CV database and career advice. The *Main* page layout is designed to try to prioritize the users' attention to the most important information. You could also put a table of contents on the first page so the user will get a perspective of the site structure.

3.1.5 Back to Forms

If a user is doing the post CV procedure the user's input should be saved in a database in the system. If the user goes back to a form the saved information is retrieved and put as default in the form. The Web browser should not cache the pages with forms so they are reloaded from the system each time the user goes back so that the system can insert the information in the form. Three techniques are combined, putting the information in the database, disabling caching of form pages and remembering the user-id in a cookie.

3.1.7 Language

A language issue that came up was whether to use terminology such as "*My Account*" referring to the user's viewpoint or "*Your Account*" referring to the system's viewpoint.

One way of doing it is to use the first alternative, the user's viewpoint, in actions the user takes such as links and buttons. Feedback is written as the Web site speaks to the user. This way the interaction would be a dialog between the user and the Web site. For example the Web site might ask *"Would you like a personal Web address (URL) for this CV?"* and the user can check a box with the label *"Yes, create a Web address for this CV"*.

When deciding what words to use Nielsen's voting technique, described in section *2.6.1 The Eight Golden Rules*, will be used. Test users may suggest words and state which words they understand and prefer. For example, should the Web site use the word CV or the word résumé?

3.1.8 Presentation of Job Postings

On the page where job postings are presented there is much information. In previous market research and interviews during usability test, people have said what they want to see in a job advertisement. How the layout will be for this page will be determined by letting potential users sort the information by how important it is for determining if a job is interesting. While doing this procedure they may also suggest more information than the information given earlier.

3.1.9 The CV Posting Process

This problem with the CV posting process is how to do this process suitable for as many users as possible. Should you do it in one step, just using one single Web page, or in several steps? Another issue is what information the users should be able to give. In earlier interviews some test users said they preferred forms and some said they preferred copy and paste. Those interviews were with students, which is not the target group for this Web site. In the conducted market research six out of seven said they preferred to copy and paste and one wanted to use forms. Most users will probably prefer to copy and paste but the option to use predefined forms should also be available.

Guidelines to consider:

1. Design dialog to yield closure, golden rule number four.
2. Users should be able to do things in a sequence they would naturally use, golden rule number seven.
3. Users should have an overview over the process.
4. Users should be able to choose alternatives.
5. Feedback should be presented after every user action.
6. Actions should be reversible, see *3.1.5 Back to Forms*.
7. Any information should only be filled out only once.
8. The process should be quick.

Alternative one is to put everything on one page. This is used on Headhunter.Net, which has put much effort on usability issues.

Figure 3.4 shows a page that uses this design. The size of the scroll bar to the right indicates that this page is quite long.

If this is used this page will include the copy and paste field for the CV and if the user wants to use a predefined form it will be available through a button, as shown in *Figure 3.4*.

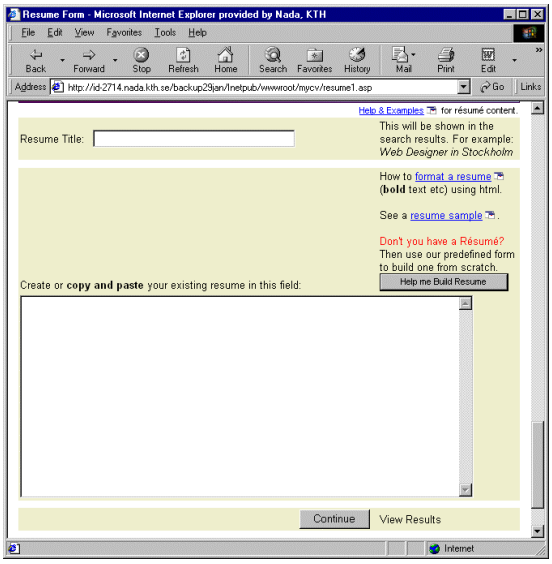

Figure 3.4 *The page with the form used for posting a CV. A text field for copy and paste is shown.*

The advantages are:

1. The user will have an overview and will not have to click back and forth between forms.
2. The CV will be shown as feedback directly after filling out the form.

Disadvantages are:

1. The page might be very long.
2. Too many fields might overwhelm the user.
3. When filling out country of residence, citizenship and in which countries the user is willing to work, the user might have to fill out the same country three times. List of countries are also very large, five kilobytes, which makes the page large. Using JavaScript may avoid this problem.
4. The user might not have the same sense of relief since there are many fields on the page. However the fields will be grouped together, which might help the user feel that something has been finished after finishing each group of fields.

Alternative two is to have two different pages. The advantages are:

1. The dialog yields closure, which makes the user feel relief after completing step one.

Disadvantages are:

1. The user may have to go back to the first page from the third page if the results were not satisfactory.
2. The user will not have total overview over which fields are available.

The input fields that are included might vary through the design process and this might affect which alternative is best. If many more fields have to be included because recruiters wish so, the best alternative might change. Nielsen recommends parallel design; that different designers design several alternatives at the same time.

After trying several different solutions it seems like the best solution is to put everything on one page. The main reasons for this is that the user will get a good overview over how much and what he or she is expected to fill out. The user will also get feedback immediately at the next page.

The main argument against this approach is users who do not have a CV yet. For them it would be better with a two-step process, where they choose whether they wish to post their CV using copy & paste or a predefined form in the first step and depending on their answer, the next step is one of the two possible options.

Since six out of seven in the marketing research preferred to copy & paste, the CV posting process is optimized for people having a CV. It will still offer the possibility for users who do not have CVs yet to create one using a predefined form, although for them it will be somewhat more complicated. They will have to click a button next to the copy & paste field. This button will take them to another form, so for them there are two steps.

3.2 Usability Testing

The first tests are normative tests. The goal is to find out how to improve the usability of the Web site, both as whole and specific details.

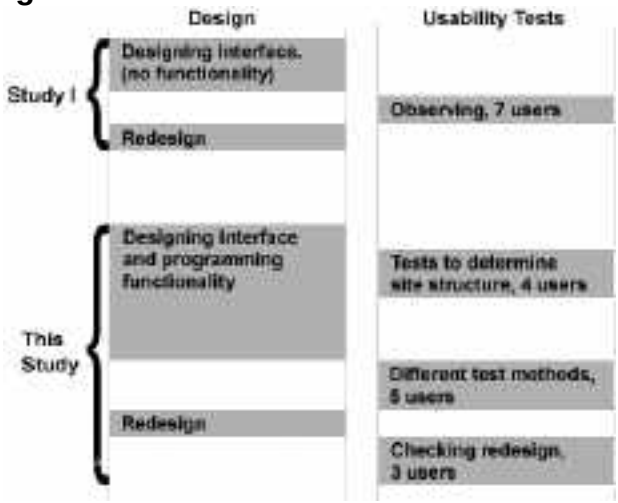

Figure 3.5 *An overview of the design process and usability tests.*

There were three sets of test, see *Figure 3.5*. The first one was done early and its purpose was to discover how to make the structure of the Web site. After the prototype for the Web site was almost finished another set of tests were made. This time different methods were used for the usability tests. A recruiter was also interviewed. The last set of tests was done to confirm or reject the solutions to the usability problems found in the second set of tests. In total there were eight test users and three of them did more than one usability test. Their backgrounds are described in the section *3.2.2 Usability Tests*.

3.2.1 Tests to Determine the Site Structure

At first the Web site was supposed to have a tree structure with menus and sub menus. When designing this structure it was discovered that users had to be involved in this process too, as it was fairly complicated. Other information that was sought was which information job seekers want in job posting and how they wish to post their CVs.

The first part was supposed to be done by showing the users a set of sketches of the pages in the site, and they were supposed to do special tasks by pointing at the sketches. Unfortunately while doing a pilot test it turned out that the user who did the test felt very uncomfortable doing this. Neither did it seem to give much valuable information. This part was skipped and instead the users received cards with page titles written on them, which they were asked to sort in a meaningful way. They were asked to sort them in piles of cards belonging together or cards they believed should be one page. Of course, all users came up with slightly different structures but a pattern was observed and that was that the users preferred to make *few* piles and combine different things to the same page. The conclusion was that they want few pages instead of many in a logical structure.

After that they were asked to sort information from job postings, such as job title, location etc by importance and to state whether the information, for example location, *had* to match their preference or if they just thought the nearer the better. Finally they sorted information in CVs in piles and commented on how they wanted to post it. These two latter parts did actually not give that much information, since there were only four users doing it and there were no obvious patterns in their behavior. It did remind the designer that different people want different things and it gave at least some clues what the users thought was important.

When doing these tests one user was also shown the prototype of the Web site and asked to try it. Some minor usability errors or annoying details were discovered, which of course was good. The next set of usability tests would go smoother and the prototype of the Web site would have less annoying details when doing the test.

When doing one of the tests, CareerBuilder [CareerBuilder] was tested for the purpose of testing a job site that was working. That Web site had a confusing structure, strange metaphors that neither the test user nor the observer understood and links to the same pages with different labels that confused the user. In general the user felt a bit confused so what was learned was more how to not design than ideas about how to design.

3.2.2 Usability Tests

There were five persons involved in the usability test. The first was working with Internet and actually interested in usability issues, the second was a law student, the third was an engineer that recently got his first job and had some experience of job sites, the fourth was a recruiter and the fifth and last was a person with Internet experience that recently got a new job. There were also two recruiters working for an Internet Consultancy and a major software company. All persons were Swedish. First of all a pilot test was done with a colleague to the user interface designer.

First the test users would get 5–10 minutes to explore the Web site. After that the test users was given the following tasks:

1. Post a CV
2. Find a job they like
3. List all jobs posted the last 30 days sorted by salary. (To do this they have to go to the *Options* page and search from the *Advanced Search* page.)

When they were done they filled out a questionnaire and were interviewed. The tests took about sixty to eighty minutes each. Dr. Jaime Sánchez at University of Chile designed the questionnaire and its focus is on usability of Web sites [Sánchez, 1999]. It was translated into English and slightly modified for these tests. The questionnaire is included as *Appendix A.*

The questionnaire that the test users filled out, starts with general subjective questions. After those questions follow about fifty questions arranged in twelve sections. *Visibility of the state of the system*, *Relation between system and real world* and *User control and freedom* are some of the sections. See *Appendix A* for all sections. It finishes with summarizing questions of these sections. The same questionnaire was used in the first study [Backlund, 2000] but with some modifications. The questions are in fact positive statements and the user chooses from *Strongly Agree*, *Agree*, *Neutral*, *Disagree* and *Strongly Disagree*. Since most statements are positive, for example "*It is easy to find information*", *Strongly Agree* is best. When calculating the grades, a scale from one to five is used, where five is best. *Strongly Agree* would thus be translated to the grade five. This is done to calculate average grades.

The questions in the interview were very general. It began with questions like what they thought about the Web site in general, what was good and bad. These three questions got the users to talk quite a lot. Depending on their answers they were asked more detailed questions. If they had not mentioned the navigation, the opening of new windows, the CV form etc, they were asked about what they thought about those things. There were about twenty questions in all.

Since there had been user tests before, the pilot test went reasonably well, some spelling errors were found and the questionnaire turned out to be quite large and time consuming. Some questions were removed and some were clarified.

The first usability test was supposed to be a retrospective test with video recording of the screen followed by a discussion while watching selected parts of the video recording. The idea, as described in *2.3.3 Retrospective Testing,* is to watch parts of the test that are difficult to understand. In this case the test user was interested in usability himself and had no problems talking out loud what he was doing and thinking. As he kept talking during the entire test it was really easy for the observer to follow and understand what happened. Because of this there were no parts the observer needed to go back to and discuss.

The second test was with two persons, doing a codiscovery learning. The test users would post one CV each and find one job each. As they did not know each other before, the test became somewhat stiff. They did cooperate some, but as they were supposed to post one CV each, both users moved away some from the keyboard and sat quiet watching as the other user posted his CV. In the beginning, when exploring the Web site freely they talked some, but they became quieter as they did their tasks. At one occasion they thought about doing a mutual CV, one user had started to fill out his CV and suddenly asked for a work experience from the other user, that is, mixing their experiences and personal information to one CV. They changed their minds as they were not encouraged, which probably was a mistake by the observer. Even though it would have been a strange situation to mix a CV, it would have encouraged the conversation, which is the idea with this kind of test. The tasks might not have been very suitable for codiscovery learning.

When the situation had become a little stiff, the observer tried to make the test less formal and suggested a break to get the conversation going. The situation did not change much even though they talked and seemed to get along during the break.

A recruiter was contacted and asked to be interviewed and to evaluate the prototype. She came from an Internet consulting company that had had a rapid expansion during several years. It was desirable with a recruiter's perspective on the prototype and not only job seekers' opinions. The interview started with a conversation about how she, as a recruiter, worked in general and how she had used job sites. After that she was asked to evaluate the prototype of the Web site.

The last test was a "*normal*" usability test, that is, a test user that is being observed while doing the tasks. As with the other tests this test was also followed by a questionnaire and an interview. The test was done with a person that did not know that the observer had designed the interface. He was also more interested in looking for jobs than for example the law student was. This made him more negative about the site and he had more critique than the others, which of course is good. To learn more about usability tests with an observer see section *2.3 Description of the Usability Tests* in the first study [Backlund, 2000].

4 Results

The results from the tests are: comments from the interviews with the test users and the recruiters, notes made while observing them, including their comments and behavior, and the questionnaires. Compared with the first study fewer usability problems were found. In this study three major errors were found and a total of 19 minor errors or 3.8 per tester. In the first study [Backlund, 2000] eight major errors were found and 26 minor or 3.7 per tester.

The comments from the users were more positive this time but the average grade in the questionnaire was 3.6 compared with 3.8 in the first study [Backlund, 2000]. This time the informal discussion after finishing the test gave less valuable insights since there were fewer errors and the observer was more experienced at this point, both in general and regarding this very Web site.

4.1 Comments from Users

Several of the users commented that it was easy to find information and that the Web site had a clear design. When the users explored the site, in the beginning of the tests, most of them saw almost the whole site in just two to three minutes. This is taken as a good result and a confirmation that it is easy to navigate it. The interface of the Web site that was tested is shown in *Figure 4.1*.

***Figure 4.1**. The main or home page before the redesign.*

One user said that the colors were "*fresh and delightful*" and the test user who was interested in usability commented that they were good for the legibility. The engineer who tested it and who had used job sites before commented that the design was clear although it did miss something in the graphical design. He was probably right since the designer is not a professional graphical designer. He also said "*It is not very messy, but maybe that will come when you have more sponsors and stuff…*". and he said it in a tone that suggested that it did not look as professional as a "*messy*" site does. This quote is quite interesting if you think about it. One user thought it looks old-fashioned. In general the users liked the design though it could need some polishing from a professional graphical designer.

Most test users said in the interview, when asked, that they liked the help window. One person discovered that the link at the bottom labeled *Help* also opens in a new window even though it does not say so. All other links that open in a new window have a small icon marking this and an alt-text that says it will open in a new browser window. He discovered this inconsistency, which shows that he had understood the idea that links that open in a new window are slightly different.

The user who had used job sites before said, with a sigh, that he "*hate these kind of forms*" when he came to the CV form.

The Internet professional said that he wanted to upload a .pdf document since then he could format it however he wanted and would not have to write it all over again. He expressed some concern about the fact that all types of CVs would have to be searchable.

The codiscovery learning test revealed one thing the others did not: when posting a CV for the first time the two test users discussed whether they needed an account or if that was created at the same time as they posted the CV. They came to the conclusion that it probably was created at the same time, which is what actually happens. The Web site ought to communicate this better to the user, by saying so clearly and perhaps changing the title for the password field from "*Password:"* to "*Choose a Password:*".

The test user who is an engineer expressed concern about the security. He asked out loud whether or not he would be logged out from his account if he clicked on "*Career Advice*" for example. When he did this, he erroneously assumed that he *had* been logged out but discovered after a few seconds that he had not. He commented that the Web site did not seem very secure.

The other test user who participated in the codiscovery learning test and who is a law student did not have a CV. When he was filling out his CV he copied a part from the CV sample and pasted it in the page that helps users build a CV. You should be aware that many users who do not have CVs might do this, and therefore there should be several CV examples with texts that are suitable as examples.

One user wanted to try to apply for a job and complained that you could not use your CV when applying for a job. This function is not implemented yet. The reason is that you should not do too much design before testing a prototype, the process is supposed to be user-oriented so it should not be tested only when all is done, but also during the process. Therefore the tests were conducted before this function had been implemented. This showed that at least one user wanted this function.

One user complained some about missing functions, he wanted to e-mail a job to a friend and he also wanted the Web site to send him new job posting per e-mail, as: *"all the cool sites do"*.

The interview with the recruiter gave some valuable feedback. Her evaluation indicated that the prototype was basically on the right track except for a few details. She talked about how recruiters work. She mentioned that they always require a CV, which they register. The company she worked for did not use job sites that much, because of lack of time. They used them if they needed a large number of CVs to look through. According to her the type of job seekers using job sites are people trying to change their profiles or want to be seen. This can be people changing career fields or people who recently graduated for example. People who are attractive on the job market seldom use them since companies contact them frequently anyway.

She had some opinions about details on the Web site, for example the list of industries that many sites use. In this prototype there is a list of categories that corresponds fairly well to a list of industries but with only three categories so far. She complained that many sites had lists where some competences or industries were missing. She also mentioned that for example copywriters do not look in the

category IT Services, where her company is listed. Her company might miss out on copywriters because of this. She suggested that if a list of categories, competences or industries is used it needs to be thought through carefully. As a recruiter she wanted the CVs on a job site to be as similar as possible to a "*real*" CV, that is, a printed CV used to apply for a job.

4.2 Questionnaire Results

The average grade in the questionnaires was 3.6 compared to 3.8 in the first study. However the users' comments both in the interview and while testing overall seemed more positive. It is still a worse grade and the interface needs to be improved. The questionnaire had been translated from Spanish to English and some questions had been removed, which might have affected the average grade. Other things that might affect the grades are the relations to the observer and the fact that the first study was done in Chile and the second in Sweden one year later. The questionnaire was in English in this study even though the users were Swedish. However it is obvious that the Web site needs to be improved and you should set a goal for grade, number of errors and for the time for certain tasks to decide whether the quality is good enough.

The grades from the questionnaires match well the observations made and what the users said in the interviews, considering things they complained about or praised, except for one or two questions. As with the first study [Backlund, 2000] the questionnaires do not give very much information, but of course questionnaires are more difficult to analyze than an interview or an observation.

Thing that are confusing with the results are the questions about errors and error prevention. Some users made comments while filling out the questionnaire that they did not see any errors and therefore chose the alternative "*Neutral*" for those questions. The user errors the Web site is prepared for is if the user ignores a mandatory field in the CV form or if the user searches for jobs and finds nothing. The second of these two errors happened a few times, because there were very few job postings in the database, but the first never occurred. Still it is not certain that the users referred to this when answering to the question but it might be that they referred to usability problems they found. For example one user had difficulties using the *Options* page combined with choosing between the *Search Jobs* page and the *Advanced Search Jobs* page, in this case it might be reasonable that he thought this was an error while answering the questions about errors and error prevention. The meaning of those questions is to evaluate the error prevention in the system and not usability problems, of course.

Topics that got positive response from the users in the questionnaire were that it is easy to navigate, it loads fast, good *aesthetics and minimalistic design* and *user control and freedom*. Topics that got negative critique in the questionnaire were *error prevention* and the expected *functionality and capacity* of the Web site.

4.3 Usability Issues and Suggested Improvements

Most of the users could explore almost the entire site in two to three minutes while doing the usability tests so the structure and the navigation seem to work well. The user also made comments confirming this.

Compared to the first study [Backlund, 2000] there were significantly less and smaller problems. In the first study there were eight major problems and 26 minor problems found. The Web site then had to be redesign from scratch.

In this study three problems considered major were found, even though they were much less catastrophic than those in the first study [Backlund, 2000].

1. **The Options page**. The *Options* page had two functions, partly to decide how to present the search results and partly to decide which country to include in the job seeker's personal list of countries. There were thus two forms on this page and some users thought it was one form and were confused about how it worked.

   To solve this problem the presentation options, including how many job posting per page, how detailed descriptions of each job posting and how to sort the job postings, will be put on the *Search* page. While looking at solutions from other Web pages, as Nielsen recommends, it was found that most sites put this kind of options on the *Search* page and some put it on the *Search Results* page. There is a risk that there will be too many fields in the *Search* page or *Search Results* page that will overwhelm the user.

2. **Unclear sign-up**. In the codiscovery learning test the users felt unsure whether posting a CV also created an account. One user was not sure whether the password he chose would work for log in.

   To make it clearer you could write a short text at the top of the CV form, which says that posting a CV will create an account. The label for the password field will be changed from "*Password*:" to "*Choose a Password*:"

3. **Too international**. The site is supposed to be international but there are no instructions about which language to use in searches and in the CV form. Addresses might vary some between countries, for example not all countries use zip codes and not all countries have states like the United States and Canada. A person in the United States will probably want to use a list of states when searching for jobs or saying where he or she wishes to work.

   A sub site for each country should be used. For CVs you can also make a solution where the user chooses the text in the header, like on Yahoo! Careers [Yahoo! Careers]. This way the users could make CVs in one language only, and not having headings in English when the rest of the CV is in another language.

The minor usability problems were:

1. **Speed**. There is still too much graphics. Pages load within eight seconds on a 56k modem but they could load much faster. In the top menu there are several images that could be removed. This was not done yet because of lack of time. In general the users were content with the speed, giving it grades from excellent to neutral.

   Using tables instead can easily speed up the pages. It might be tricky to make pages look similar in all browsers and the design might be less appealing, since you cannot make round shapes without graphics, but if a graphics designer works with this it should not be a problem. The main goal for the Web site is not becoming beautiful, it is more important that pages load fast. Hotmail [Hotmail.com] is an excellent example of minimalistic design.

2. **Advanced or Basic Search**. One user was annoyed by having to click between these pages several times. He started on the *Advanced Search* page and clicked to the *Options* page and from there he clicked on a link to the *Search* page, instead of using the browser's back button. He ended up on the *Basic Search* page. So to get to the *Advanced Search* page, as he wanted, he had to click one extra time, which annoyed him. Since this procedure was repeated several times he got really annoyed.

   By making these pages into one page and using cookies or session variables, a smarter version of cookies, it can be decided which version to display. If the user was using the advanced version he will see this the next time he visits the page.

3. **The country list**. Instead of having one very long list of all countries, the user may define a personal list of the countries that interests him. Normally this list would be very short, perhaps only one country. A short list would download faster and the user can scroll a short list faster than a long list. If the user is on a domain in Sweden for example his list will include Sweden automatically. Now, the problem is that some users thought that by adding a country to this list, all searches would automatically search for job postings in all the countries on the list. There is no option for this kind of search.

   A solution is to explain the list better on the page where users add countries and add an option labeled *"All countries on my list"*.

4. **Details in the CV form**. The following details were found in the CV form:

   1. One question is whether the user wants a personal web address for his CV. This can be used for linking and to give to recruiters that are not registered users on the Web site. The question uses the word *"personal web page"* when what is really meant is *"personal web address (URL)"*.

   2. The dropdown-list labeled *"length of your education"* is unclear. Two users thought they should include elementary school in this, which is not the idea.

      It should be clear that this refers to university studies.

   3. The form includes a question about number of years of work experience in the field the job seeker is aiming at. According to the recruiter it is more interesting, both for recruiters and job seekers, to state total number of years of work experience. It might be interesting with experience related to the field of interest as well so perhaps this should be included.

   4. The list of countries where the job seeker has legal rights to work will be irrelevant for most job seekers since most want jobs in their own countries.

The recruiter suggested a question where the job seeker is willing to work with follow-up question whether the job seeker has legal rights to work there. If this follow-up question is included the job seeker can only choose one country.

5. The order of the fields is confusing. There is a question about job status between two questions about the location where the job seeker wish to work. If there are more than one question about the location they should be close together.

6. The fields asking for years have no specification whether there should be two or four digits for years. The Web site allows both types and it is impossible to write more than four digits but one user had to ask about this.

   There should be some text about this, for example "*YY or YYYY*" next to fields where years are to be filled out. Some job sites use a dropdown-list for this, listing all years from 1950 or so until 2001. These lists prevent the user from wondering but he cannot choose how to write it. Such lists are about one kilobyte large, which would make the form to build a CV about ten kilobyte bigger.

5. **Inconsistent help link**. The help link at the bottom of all pages opens in a new window even though it does not specifically state this.

   One could mark this clearly and always have help in a new window. You could also build an additional set of help pages, with almost the same contents, for the same browser window, which would be a small inconsistency but you would gain the possibility to link from the help text to pages and also let users who prefer one window use this.

6. **Job recommendations**. The job recommendations on the first page were meant to be a marketing instrument for positioning the Web site as having high quality jobs. Two users commented on this, one said that no one will gain anything from those and the other said, "*There are recommendations even though they do not know anything about me, that's how good this site is…*"

   The job recommendations should either be removed completely or have another title, for example "*Most Popular Jobs*".

7. **Browsing for jobs**. The browse for jobs function is not good enough. It is not very visible and more types of browsing are needed, depending on what the job seekers like to browse by.

   There should be a better category list and maybe a page with links to different categories, industries or types of professions. A good interface for browsing is the type Yahoo! and several other directories use.

There were about 19 small usability problems in total and about six bugs.

While the users were doing the tasks the observer made notes about how long time they took. Although these times are not very exact, it took 9–18 minutes to post a CV for user without a CV and about 11 minutes for users who copied & pasted their existing CVs. It took about 3–7 minutes to list all jobs posted the last thirty days sorted by salary, which was the third task in the tests.

To educate the users about the site, for example answering such questions that could be in *Frequently Asked Questions* section one can implement a *"Tip of the day"* function. This would be a little square at the bottom of some or all pages giving tips and facts about the site so that user would learn more and more about the Web site even without going to the help pages. It is a little like the idea with examples that change in the *Search* page, those examples get more and more advanced so beginners get easy examples and experienced users get more advanced examples. Two other improvements are including a feedback page and making the sitemap more like the graphical design. The sitemap ought to help the user build a mental model of how the site is structured. By putting *Search Jobs*, *My Account* and *Career Advice* in the same positions as in the menu, the sitemap would be easier and help the users build such a mental model.

4.4 Redesign and Verification of Solutions

After implementing solutions to the problems found, those solutions had to be checked to see if any improvements had been made. This time, as the test would not be so extensive, almost no programming was made. It was decided that the browser compatibility had to wait until the interface was absolutely finished, so the interface was designed to work best with Internet Explorer and when it is done it will be adjusted to fit Netscape as well, although it currently works fine with both browsers.

Two of the users from the first test did a second test and one new recruiter analyzed the interface. The tasks were the same for the users and there was an informal discussion about the interface, but the users did not fill out a questionnaire this time.

This time some of the problems were gone, some still needed improvement and one new problem was found. The two users who had used the Web site before seemed to remember how to use the Web site and in general seemed to be used to the interface, which is positive.

The user understood how to use country list and the option "*All in my list*", that is, they understood that it meant all countries they had added to their list. The use of the *Search* page, went smoother, they did not have any problems with choosing *Basic* or *Advanced Search*. The solution that was implemented was that the system remembers which type of search the user used the last time. The solution does currently not support bookmarks, but it can easily be implemented. See the page in *Figure 4.2.*

Two new usability problems were found. The first one was that the user complained that his input in the advanced search form, see *Figure 4.3*, still disappeared if he clicked to go to another page. In this particular case it was a usability problem for the browser since he had not done any searching yet and therefore his input was never received by the system. It showed, however, that the checkbox for saving did not work well. He never saw it. Either it should be checked by default or all inputs should be saved during the entire session by default. Probably the first alternative is the best since it gives the control to the user.

Figure 4.2 *The search page with the basic search form. The three buttons on the top of the page are not images but are made with tables and use JavaScript to change colors when the user moves the cursor over one of them. The interface has four images in general and this specific page contains three more for the heading.*

There was some uncertainty about how to design the *Post CV* page. When testing it, it worked fine and all users who needed to do so, used the button to build a CV from a predefined form.

The page was improved some in its visibility. There are now two alternatives of how to post a CV, copy & paste or build one from scratch. In the second usability test all the users understood these two alternatives and that they had to choose one of them, see *Figure 4.5*.

Figure 4.3 *The search page with the advanced search form. The page is exactly the same as the basic search, except that it has more search fields. In this image an alt-text is shown for the keywords field.*

However one thing was unclear and that was when one of the users edited his CV and tried changing between the two alternatives. He was not sure whether there were two versions of his CV or only the type he used latest. To solve this one could either change the interface of the edit page or split the CV posting form to two pages where the user chooses one alternative on the first page.

The second problem was that the help boxes to the right do not seem to belong to with the input fields in the *Search* page. The same kind of help boxes works better in the form for the CV, since there the input field *and* the help text are in the same square. In the *Search* page the square is to the right of the field and at least two persons in the last usability test did not understand that they belonged together.

Figure 4.4 *Top of the post CV page.*

One of them was the second recruiter who, when evaluating the interface asked how you know what to search for in the key-words field. The form on the *Search* page should have a similar design as the form on the *Post CV* page, see *Figure 4.2*, *4.3* and *4.4*.

Figure 4.5 *Bottom of the post CV page.*

The recruiter was interviewed and showed how she uses job sites. The company she worked for, a major software company, had a job section of their own and also used some job sites. After the interview she evaluated the interface. She did not have any major complaints but said that the job status that job seekers state is not of any interest to her. She also said that it is good if the job seekers may state where they wish to work and that she thinks that they will probably want to search by type of profession or some kind of category.

During one of the usability tests a competitive job site was tested as well, namely 6figurejobs.com [6figurejobs.com]. The results from this test showed that this particular user liked searching by industry and job function, which was a list of professions more or less. The *Search Results* page worked badly for this site, it disappeared from the browser's cache memory every time so the user had to click back one more step to the search form and had to search again.

5 Discussion

The results from the test showed that the interface for the Web site had improved slightly since the first study [Backlund, 2000], see *5.4 Comparison of Results from the First Study and this Study* for more about this. The questionnaire result was worse and the minor usability problems were almost as many as in the first study but users' comments were more positive and the major usability problems fewer.

Only one person was involved in the design process of this Web site. That person had to work as interface designer and perform the usability tests, that is, be the observer *and* do the interviews. This is in general not recommended, unless that person can accept negative critique of his own work and is aware of the fact that it is a bias that the same person designs the interface and later serves as observer and interviewer when testing the interface. Although, it has to be said that, as an interface designer you can gain lots of knowledge on this.

In this case the interface designer was not only observer and interviewer, but also programmer, graphical designer, copywriter, marketer etc. Copywriting and graphical design is probably what the Web site lacks most. However, when working the biggest problem was combining the two roles as interface designer and programmer. This is because as an interface designer you want to try out different approaches, of which some may require much programming, or useful features. As a programmer you do not wish to work a lot for something that might not be used in the final solution, especially if programming is not your main interest.

Combining these two roles is difficult, as designer you want to try different approaches and features but since you know that you must implement all this as well it restricts the work of the interface designer. This means that all solutions that are complicated to implement technically, even though they might be great for the usability, will make the interface designer think twice. During this master's project the interface designer *did* think twice several times but tried to do the difficult work if it seemed to be worth it. If one person is both interface designer and programmer there is a risk that this person produces a worse result for the interface, being aware or unaware of this.

A general recommendation is to do as little programming as possible until it has been decided how the interface should be.

During the studies performed, it went fairly well combining the roles as interface designer and observer and interviewer. As the interface designer was eager to learn he was glad when users found things to improve and did not take it personally, at least it was not perceived that way. At some occasions when some users acted like expert interface evaluators, their critique could be a little provocative, but on the other hand it was interesting and nice that they were interested enough to make an effort to suggest new solutions. You have to remember that it is more important how users behave than what they say. When a test user suggests a new solution, the observer should at least note that the user was not satisfied with something. You should always try to be as objective as possible.

There were differences when doing the test with someone who knew that the observer had designed the interface that was evaluated and with persons who did not know this. Test users who were unaware of this gave more honest feedback. Whether or not the test user knew the observer personally also affected their feedback. The most important characteristic for test users were whether or not they were interested in job seeking, for example someone who was interested in his or her career or recently changed jobs gave much better feedback than for example a student. This criterion seemed more important than choosing test users from certain

professions that the Web site aims for, in this case Internet professionals, legal professionals and MBA graduates.

5.1 The Methods

This study included three different types of test: normal observations where users are encouraged to talk out loud, codiscovery learning, and retrospective testing.

The codiscovery learning turned out to be a disappointment. The reason for this was that the two test users did not know each other before the test. Therefore the test became very stiff. It still gave some feedback that did not come up in the other test, so with users who know each other it might be a very good idea.

The retrospective test went so well, that it was never necessary to watch the videotape. Still, if the user feels comfortable with a video camera, the observer can use the recording to refresh his or her memory. In cases where a complex structure of a Web site or interface is tested this method might give more.

The main reason the retrospective test turned out to be a normal test with an observer was that the test user talked out loud during the entire test about everything he did. If the observer can induce the test user to do this, talking out loud is an excellent method for usability testing. For this set of test, the test where the user talked out loud during the entire test gave superior results. It was a lot better than the codiscovery learning test, even though such a test might be better with two friends or colleagues doing it instead of two persons not knowing each other.

For an interface designer or usability test observer that is not very experienced an informal chat with the test user after the test might teach him a lot both in general about usability and interfaces but also about the specific interface being tested. For more experienced interface designers and usability test observers it may also give interesting feedback. The informal chats with test users in the first study [Backlund, 2000] gave much more knowledge than in this study, which is quite natural considering the difference in experience.

In both this study and in the first study [Backlund, 2000] it was found that questionnaires, interviews and observations give different types of feedback. Observing gives interface details, interviews give more information about utility and what the users want and questionnaire give concrete numbers about how pleasant the Web site is to use.

5.2 More Functions for the Web Site

The overall goal is to make the Web site as useful as possible. Usefulness is both utility and usability, which is the main focus of this report. To increase the utility of the Web site, that is, to add more functionality that is needed, the following additions will be considered:

1. An apply-to-job function, where the users can send e-mails to a company and include their CVs from the database.
2. The possibility to choose graphical design of their CVs. The Web site would offer several different design templates that the users would be able to choose from.
3. Jobs per e-mail to job seekers. Many job sites have this and use metaphors like *"job agent"* or similar. This service would simply be called *"jobs per e-mail service"*, that is, not using a metaphor. Users would be able to set

searches and control how and when e-mails with newly posted jobs would be sent to them.
4. The possibility to save job postings in a list on their accounts for later use.
5. Career advice and tutorials. Sample CVs would add both utility and might attract visitors from search engines.
6. The possibility to post more than one CV. One user requested this and some job sites support this.

For one person to make an entire site is much work, especially since several competences are needed and since this master's project comprised only a few months, not a year or more. Because of this not all desirable things can be included within this master's project.

5.3 Future Improvements

The next thing to be done to increase the usefulness of the Web site is to do test on competitive sites that seem to be useful. This will give more information about how users wish to search and browse. The job browsing on the Web site has to be improved significantly. You could design a page for browsing and add lists of categories or industries in the *Search* page. This list has to be carefully chosen so it fits both companies and job seekers. To make sure that as many job seekers and companies can find at least one appropriate category, the list could contain both categories and subcategories. For example *Management* with subcategories such as *MBAs*, *Managers* and *project leaders*, to mention a few. Most job sites just have one level with detailed categories.

To test the competitors would be a good way to do a task analysis, *what* the job seekers wish to do on a job site. This is the biggest weakness of this study and could be made more certain with such tests. Then the functions in the list in the previous section can be implemented or ignored.

A professional graphics designer should design a clearer graphical profile. The profile should have a minimalistic design as it has now. It should also allow more consistent design and make use of the entire width of the browser window. As it is now it uses a width of 619 pixels that can be viewed on all computers, but on bigger screens there are lots of white space that is unused. Pages should be designed to use 100 % of the width of the browser window, but be perfectly viewable on a 640 pixels wide screen. Then no one needs to scroll horizontally and users with bigger screen can make use of the screen area.

5.4 Comparison of Results from the First Study and this Study

In the first study [Backlund, 2000] the prototype of the Web site that was tested only had an interface and lacked programming and databases, which had been implemented in this study. Therefore this study probably gave more realistic results. The negative side was that much effort was put into programming issues and that during the usability test some bugs were found in the program code.

In this study users were involved in making the structure of the Web site, which was not done in the first study [Backlund, 2000]. The usability tests went more smoothly and unlike the first study no one postponed or canceled a test. The users were better informed when asked to participate, which might be the reason for this. The test users were more appropriate than in the first study. In the first study the majority

was students and in this study only one user was a student. Recruiters were also involved in this study.

All the findings in the first study [Backlund, 2000] were not implemented, some because the master's project had to be limited and some because better solutions were found. A few of the usability problems in the first study were misinterpreted. One example of this is that the test users in the first study had problems filling out their addresses since the fields were not adjusted to the Chilean address format, for example zip codes are not used in Chile. The solution to this was to make the form for addresses very open, with three fields for the address, one for each line. The problem was in fact more complicated than that. As suggested in the results section of this report the Web site has been designed to be too international and general. It should be adjusted to fit separate countries.

The usability problems found in the two studies are:

Major Usability Problems in The First Study

1. Speed
2. Lack of instructions
3. The sub menu
4. Lack of feedback
5. Frames
6. Consistency
7. No main page
8. Visibility of the state of the system

Major Usability Problems in this Study

1. The *Options* page
2. Unclear sign-up
3. Too international.

Some of the problems found in the first study [Backlund, 2000] were not implemented perfectly in this study, for example the lack of user instructions since all details on the Web site may need a corresponding instruction. However a consistent system for instructions was designed where all input fields have a square to the left with space for instructive text.

The number of major usability problems was improved significantly but there were still 19 minor errors compared to 26 minor errors in the first study [Backlund, 2000]. This could be because the site was totally redesigned and because this time the prototype was working with a real database and not only an interface.

The process was more effective in this study. Cascading style sheets and common files for the interface were used, letting the designer concentrate more on the interface and not having to update the same things in several files.

During the studies a lot about interfaces and usability was learned. It was really inspiring to do the studies and lots of ideas were suggested. Examples of this are the ideas from the first study [Backlund, 2000] to do usability test with competitive job sites and to save user errors in a database to better understand what users are having trouble with. Funnily enough many the ideas suggested in the first study, when the author had less knowledge in usability, were found in the literature when doing the literature study for this master's project.

6 References

Books

Fleming, J. 1998. *Web Navigation – Design the User Experience*.
O'Reilly. ISBN1-56592-351-0

Nielsen, J. 1993. *Usability Engineering*.
New York: Academic Press Professional. ISBN 0-12-518406-9

Shneiderman, B. 1998. *Designing the User Interface – Strategies for Effective Human-Computer Interaction Third Edition*.
Addison Wesley. ISBN 0-201-69497-2

Questionnaire

Sánchez, J. 1999. *Usability of Educational Web Sites.*

Papers

Backlund, J. 2000. *The Making of a Usable Web site*.
http://www.e.kth.se/~e96_jba/usableweb/Report16.doc

Retting, M. 1991. *Nobody Reads Documentation*, Communications of the ACM

Root, R. W. & Draper, S. 1983. *Questionnaires as a Software Evaluation Tool*, Proc. ACM CHI '83 Conf.

Prototypes of the Web Site

Interface in the first study
http://www.c5.cl/~jbacklun
2000-02-01

Interface in the first study - Redesigned
http://www.c5.cl/~jbacklun/new/design
2000-07-01

Prototype in this master's project
http//id-2714.nada.kth.se
2001-01-31

Web Sites

6figurejobs.com
http//www.6figurejobs.com
2001-01-31

CareerBuilder
http://www.careerbuilder.com
2000-11-15

Ease of Use
http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/6
2000-11-15

Hotmail
http://www.hotmail.com
2001-01-31

Headhunter.net
http://www.headhunter.net
2000-12-01

Nielsen, J. 1999a. *Top Ten New Mistakes in Web Design*.
http://www.useit.com/alertbox/990530.html
2000-11-15

Nielsen, J. 1999b. *Ten Good Deeds in Web Design*.
http://www.useit.com/alertbox/991003.html
2000-11-15

Webmonkey
http://www.webmonkey.com
2000-11-15

Yahoo! Careers
http://careers.yahoo.com
2000-12-01