Prashamsa Mishra

User Interface Design

for existing System Monitoring Application

Helsinki Metropolia University of Applied Sciences

Bachelor’s Degree

Media Engineering

Thesis

Date 14 April 2013

Helsinki
Metropolia
University of Applied Sciences

Abstract

Author(s) Title Number of Pages Date	Prashamsa Mishra User interface design for existing system monitoring application 56 pages 14 April 2013
Degree	Bachelor of Engineering
Degree Programme	Media Engineering
Specialisation option	Audio Visual Technology and .NET Technology
Instructor(s)	Lassi Suoknuuti, Project Manager Kari Aaltonen, Senior Lecturer

The main purpose of the project was to make use of elements of interface design to create an application. Another purpose was to see how Enoro (customer) Generis system (customer's internal system) merges with the web in particular application. The goal was to create an application web interface for existing System Monitoring application.

ASP.NET framework with C# programming language, Enoro Generis System and user interface design elements were used for creating the application. The application serves Enoro's customers to monitor their servers and view the result via web interface. The application development is still ongoing. The final product was not aimed for this study but the thesis application well demonstrates the concept of final application.

The application carries good importance to both Enoro and its customers regarding business values and system robustness monitoring and immediate problem fixing. The application also demonstrates how user interface design influences application's use. The concept of the application itself is important for users who can easily know what is going on in their systems regardless of their location and the platform they use.

Keywords	user interface design, interface design elements, web security, usability

Contents

Glossary / Abbreviations

Extensible Markup Language (XML)

Markup language that defines rules for encoding document reable by both machines and human form

Application Programming Interface (API)

A protocol intended to be used as an interface by software components to communicate with each other

Windows Communication Foundation (WCF)

Set of APIs in .NET framework for building service oriented applications

Generis Enoro's platform for data management and processing

File Transfer Protocol (FTP)

Standard network protocol used for transferring files from one host to another

Info Flow Manager Jobs (IFM Jobs)

Enoro's Generis system's jobs capable of performing different tasks such as reading a file's presence on a certain folder, executing a function on scheduled time and so on.

Time Series Product (TSP)

Generis objects

Time Series Product Instances (TSPI)

Generis object instances

Label Plain text

1 Introduction

The thesis project is focused on designing a web interface for existing System Monitoring application. The application monitors servers and sends results to respective users via email. The main idea of thesis application is to provide users with usable web interface for browsing the results. The report provides general description of interface design and example application.

The work is done for Enoro whose main business area is energy data management. The company is a newly found company with integration with other two different companies located at different countries. The Finland branch, previously known as Process Vision, is twenty years old and is serving its customers well. The thesis' attempt is to provide the Enoro's customers a feature to monitor their system in an easy and effective way.

During the design and implementation of the application, user interface design elements will be considered. All study done to seek out possibility to make a usable design will be put into application development. The project will also demonstrate the process of interface design.

The subject area chosen for this topic is very big. There is no one right way to do things. This area has been in consideration for a long time now yet many web designs are failing to provide good user experiences. The goal of the thesis is to create a simple and usable web application and provide users just what they want.

Enoro internal platform, Generis, will be used to create the application which will be later replicated to web. The application will show how Generis system and web aspect merge well with each other. The application will help users use Generis without having to be in the Generis system itself making it platform independent.

2 User Interface Design

With time and creativity, developments have been part of technology. With creativity comes advancement and with advancement comes complexity but we still prefer things to be simple yet advanced. This is when we need to think how to simplify things. We need simplicity in product development and we need simplicity in product use. We need something to fulfill this need. User interface design just fits in there. It helps developer/designer to create end product in logical and sequential way. User Interface (UI) design has four main elements: Usability, Visualization, Functionality and Accessibility [2]. They are explained in section – Elements of Interface Design.

Design elements can come in use only after we analyze our potential users. If we do not know who is going to use the system, we do not know what kind of system we should make. Interfaces should always consider end users' need and fulfill their demands. In the section below, chapter 2.1, are some user experience design considerations to be made for better UI design [4;5].

2.1 User Interface design considerations

For making user experience enjoyable there are some things that should be considered before jumping into the development process. This will save developer's time and also designer's work will go smoothly. The end result will be usable.

Consistency

Users do not like inconsistent pages. Inconsistency makes things complex while consistency provides clarity. Some basic elements of an application user interface that designer should be consistent with are color scheme, style, borders, type and fonts, size, background images and effects [16].

Wise choices should be made matching the theme of one’s to be designed interfaces. For example, if we are designing pages for a coffee shop, use of coffee brown color in the background might be a good choice instead of using unnecessary bright pink which would be more favorable in soft toys selling site for little girls. Choices of these elements are very subjective but can be measured and defined by getting enough information on end users will give an idea on what UI patterns to use and how to.

Consistency also means doing similar tasks in a similar way, for example clicking on home button or the company’s logo in the banner should always take the user to the home page at all times. Consistent page layout should contain for example header, footer and body. No matter in what page of the site we are in, we will find these elements right there.

Consistency helps users remember one’s design. It adds the right feel for users to be there. Throw users different typography in different pages, different sizes and colors without any meaning or reason behind, they are going to get bored and never return to that site again. Consistency can be achieved by a thoughtful design, potential end users’ research, using master pages and CSS while developing [5].

Example of consistency:
The figure below, Figure 1 Consistent User Interface of Microsoft Office Package, shows the clear concept of consistency. Making a UI consistent helps user get familiar with system quickly.

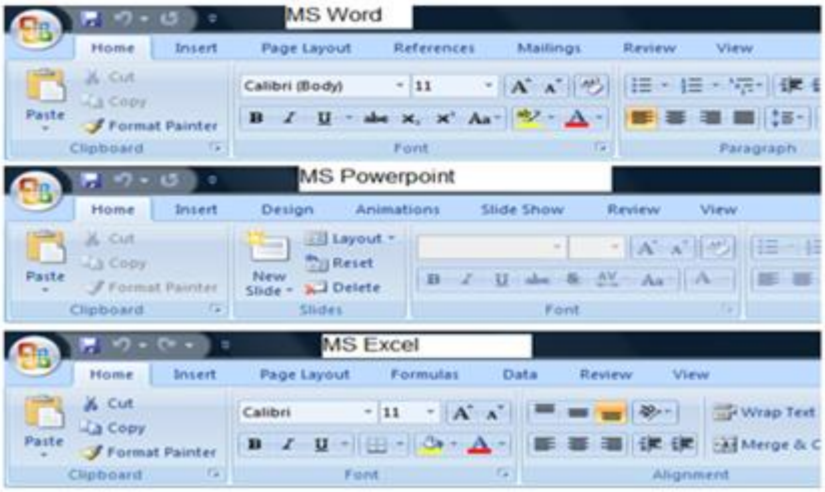


Figure 1 Consistent User Interface of Microsoft Office Package

The figure above describes how consistency is maintained in looks and features of Microsoft-office package. Once the user learns one of the package applications, it becomes easy for him to learn the rest as well. He will not have to learn a whole new interface again; he can just learn features meant for a specific application since the rest is something he has already learnt.

Responsiveness

The user might find it very annoying to keep waiting in a site and wonder how to react to a certain situation. For example, when user submits his information via registration form, he expects some kind of response from the site that says registration failed or was successful and navigates to you to a certain page. Instead if he receives a blank html page with no going back option or no any information of the registration process, it will have a negative impact on the user's future usage of the site.

The basic idea of the site being responsive is the site responding to users actions. It also means giving the users the feeling that we are listening to them. Nobody likes to talk to a tree. If the page is taking time to load, one can provide some visual graphical representation or any plain text that suggests that the page is loading or telling them the progress status. Use of heavy graphics can add to the response time of a web page. Therefore, minimizing sizes and use of them as little as possible can help minimize the problem. Using alternatives also help, for example, if an image has not loaded for some reason, alternative text stating what image it is, helps the user to at least understand the idea behind the problem. [5]

Familiar Metaphors

Use of terms already familiar to users from other existing internet sites helps them to familiarize with the website faster [5]. For example, words such as products, home (or desktop in operating systems) and shopping cart are very common to ecommerce sites. At the same time, in general sites, words such as share, signup and login are very common, so using them to the newly designed pages will help users to skip understanding these parts of the site which will help in minimizing the learning process [5]. One should always be sure to use metaphors taking cultural boundaries into account [4]. It is easy to get people offended when cultural differences are not well understood. Background research and consult with respective representative should be made to avoid possible tension and misunderstandings.

Streamlining the Experience

Streamlining the experience is about improving the navigation of a site, making navigation more consistent and enjoyable. Use of unnecessary pages or navigation paths should be avoided. The site should be kept simple and navigation clear. Top level navigation flow should be predefined. Every page could for instance have a banner that on clicking takes the user to the first page or commonly known as home page. Also, all pages should contain link to company's contact information. [5]

Along with the navigation links, streaming also deals with for example minimal usage of mouse. Users might not want to use the mouse when they are signing in to the website. Setting focus username field of the form or setting focus on submit button instead of cancel button so that user can login just by clicking enter are some examples of how users can get easily through the sign in phase. There are always simple ways as mentioned above that might make user experience better. [5]

There are other simple ways of making the user experience better. Let us consider an example of a registration form in a site. If the form has numerous unnecessary mandatory fields irrelevant to the site's purpose, user might not want to fill it up and register. Minimizing the fields according to need and providing an option to whether or not to fill in certain fields will get users to feel respected and they might eventually register. If there is no special need of having different levels of users or whole logging in process, the registration process should be made optional. Users do not like to be bothered unnecessarily. Any unwanted thing should be thrown away. Keeping them does not make the interface or the process any better.

Anticipating Usability Issues

There is no proof that users will follow a straight path when using a site. Even if they do, they might come across many issues that will leave them angry and frustrated at the site design. One very common example would be filling up shipping information or registering for a site and accidentally navigating to some wrong page and having to enter all details again when coming back to the page. This is really frustrating as the forms will have many fields to fill in. This particular issue can be controlled by for example alerting users that they are leaving the form or prompting them to save their details before leaving the form. Also saving user details for a short period of time (as there might be security issues for saving the data for longer time) and providing auto fill option might be helpful. [5;20]

Designers and developers might come across many reported usability issues after the site launch if the home work is not done properly. If the design work is being done for certain target group, research on the targeted users will help solve this problem and if the design work is being done for general users, research on the products will help. The goal of the design should never be forgotten and focus should always be kept on target.

Users should always be able to undo the operation [4;20]. They might just mistakenly perform a task. They should be given a chance and should be allowed to cancel their operation. For example, the users might accidentally add a product to their shopping cart or after putting the product to the cart decide they do not want it anymore. The users should be able to delete the product from the cart. This helps them feel secure and not forced to buy any product if they do not want to.

It is not just enough to think of usability issues that might occur, it is equally important to direct them to right direction in such issues and respond to their mistakes accordingly. This concerns error handling or error responding. In earlier paragraphs we discussed that if user makes an error, he should be helped to overcome the problem. Here we discuss the error messages that a user might get such as inputting wrong username that does not match the validity criteria. Also, the page might have taken more than allocated time for it to load. Error messages should follow simple criteria. They should be polite; users do not like impolite messages [4]. They should also be short, meaningful, positive, understandable by the user – geeky terms could be way out of user’s mind, and the messages should also be consistent [4].

The error messages help user to understand what he or she did wrong and try to correct it the next time. For example, when user inputs a username that has already been taken, the system immediately gives an error message saying something like ‘This username has already been taken, please enter another one’. This will make user reconsider the username. Error messages can be supported with suggestions. An error message could be something like ‘Username elephant has already been taken, please enter another one – some available usernames are elephant001, elep4567hant, ele2390’. Some help links can also be put along with error messages wherever needed.

Limiting the choices

Users do not like to think on how to do things [24,15], designers should help them choose. The design should be created so that people understand right away what they are supposed to do with the button or search field or any other option. Having a lot of options to go through will only de-motivate users to use the application or site as they cause distraction. A lot of information is not needed to make a good design, the less the better. Things should always be kept simple and unnecessary things should always be omitted. Hiding is another option to keep the necessary feature there yet making the interface simple for users to use it. One good example is the synonyms suggestion feature by MS - Word. It is available but it does not interfere to the users’ work every time by prompting suggestions. Whenever user wants, he can use the feature. [5;18;19]

When design work is being done, designers come across a thousand of possibilities. It is not possible to please every single user that visits the site [19]. One can provide some options to ease the usage process to some users but one cannot put in a lot of features suitable for all possible users. This is where possible end users research comes in handy. For example, when design work is being done for an ecommerce site, we cannot make it usable by users of all age groups. We need to find out what age group is an active internet user group and what group includes ecommerce site active users. Considering adults being the target group, the main focus of the application development should be on them. They should be further limited to normal groups for example adults who do not have any near vision problem. The features should be more inclined towards the former. Extra things can be thought of for latter groups like including a simple feature of making the font size bigger or a small graphical representation that suggests this feature can be put on noticeable part of each page of the site.

The default layouts and design should be always meant for targeted group whenever it is possible. If not, at least they should be product centered. Target centered designs are always better options.

2.2 User Interface Design Process

Knowing what user interface design is and what key things should be considered to get the designing work for the users, it is time to think about the process of the user interface design, how it is carried out and what steps are taken to complete it. Time management is very important for getting the project done. Making the process chart helps to keep with design and implementation deadline as it becomes helpful to track down the step one is at. Below is a chart, Figure 2 User interface design process, showing user interface design process.

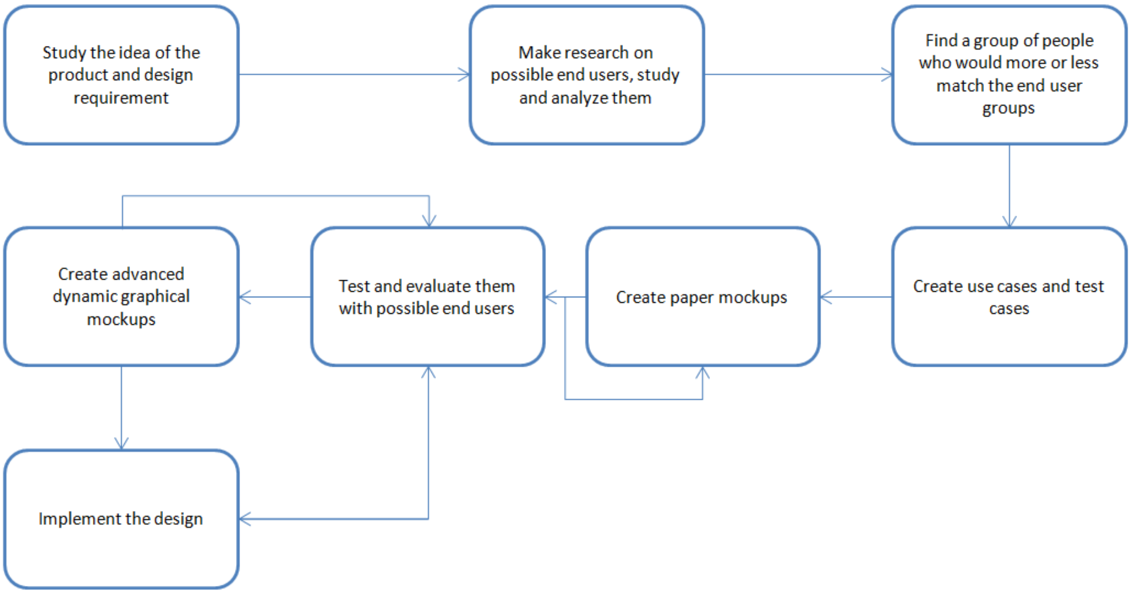

Study the idea of the product and design requirement
Make research on possible end users, study and analyze them
Find a group of people who would more or less match the end user groups
Create advanced dynamic graphical mockups
Test and evaluate them with possible end users
Create paper mockups
Create use cases and test cases
Implement the design

Figure 2 User Interface design process [Chart idea: 4;20]

The figure above mentions the basic procedural steps of user interface design. The steps are described briefly below as a process. Descriptive information on most of the topics can be found under Elements of Interface Design and Testing.

1. Study the idea of the products and design requirement
   The main thing to do when designing an application or a system with specific product is to understand the product itself, what it is and what its logic is, why people would need it. At the same time it is very important to understand the concept of the requirement, what it suggests, what it means and what it demands. It is important to understand why designing for particular system is important. If the basic concept is not clear, the whole process will be a mess difficult to clean up.

2. Make research on possible end users, study and analyze them
   Next step is to understand who the end user is. This information is needed because the design is done for the particular group of people and their interest is what keeps the design ongoing. Finding what end users want is very important. It gives us idea and taste on what kind of features do the end users want, what kind of layout do they like, what colors do they prefer, what is the level of their computer / internet knowledge and so on. Information on such little things helps designers to match the expectation of the end users. This will help to get a direction and not get lost in the design process.

It is important to know what users expect from a system. To understand user's inner thoughts and ideas, it is best to analyze what they do by observing them or questioning them [20]. The user group can range from children to old people, computer pro user to basic tasks performers, people with different personalities, people with disabilities, people from different cultures [26, 28-37]. All of them should be taken into account during the designing process. Conducting interviews with them is a way to get a whole lot of required information. The only thing to be noted while making interviews is questions should be short and meaningful so that end users do not get bored. For carrying out interviews and testing purpose, end users are needed, that is what next step deals with.

3. Find a group of people matching end users

   After successfully finding out what kind of people are using the system or application, close analysis is required to understand what they really want. A group of potential end users 5-10 (more or less in numbers) depending upon the size of the application should be found and requested to give some time to take part in end user analysis process and to help evaluate the design work.

   Approach to potential end users should be made and number of meetings should be stated beforehand to keep things clean and clear. For this, planning should be done before asking the potential end users. People do not like to be bugged, so meetings should be kept short and consistent. Questions should not be asked that could bother them or harm them mentally. Promise users some kind of reward for participating in the process and of course keep the promise after the task has been done. This will help in motivating the users to participate and remain motivated to pass the whole process.

4. Create use cases and test cases

   It is essential to create use cases for the application and based on that create test cases [4]. This will help in creating test information for the end users to test later after the design has started. Creating unnecessary cases is waste of time so only needed exact and well defined cases should be created, good enough to make the application bug proof. Cases can be added or reduced in time according to need.

5. Create paper mockups

   After the design need is well understood, end users representatives are set, they are observed and analyzed and required use cases and test cases are written, it is time to create a first version of mockup, paper mockups. Basic idea of application is drawn on a paper. Advanced idea is drawn. More features are added and asked for end users view.

6. Test and evaluate them with possible end users

   After the mockups are ready, end users are asked to go through the basic idea of the application. If they understand it, step 5 is repeated for advanced paper mockups with more features put and more navigation put in there. Simple direction like go to products page are suggested to end users and it is observed if they manage to get there. With more advanced designs, more complex tests should be done. Test cases should be brought into use to test complex scenarios which describes what to do and what is the expected result. It is always better to confirm with end users if some changes is needed to be done than to throw away all the work if users find it very complicated or do not like it at all. Questionnaire or video recordings can be used to save the collected information [20].

   No user is smaller than the other so, all users' feedbacks and comments are equally valuable, be the user in minority or majority [4]. Making a usable application is not an easy task. The only thing that assures a usable application is end users participation. All features that are designed for specific users should be tested by those specific users. No user should be left out or their thoughts should not be discarded no matter how simple the idea is as long as they represent their group of users.

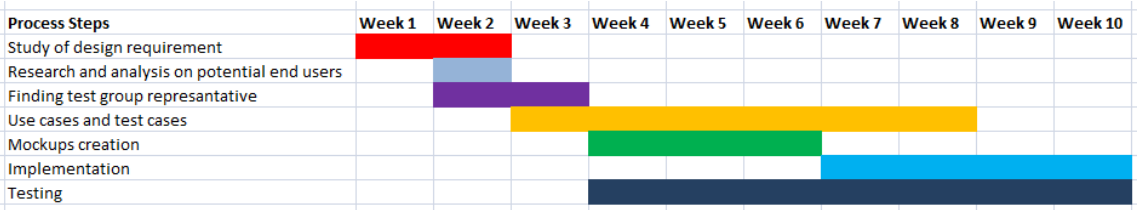
7. Create advanced dynamic graphical mockups

   After green lights to the paper mockups, mockup design should take better forms. Mockups can be created in graphical format example in Photoshop to give the real feel of the end product. They can also be created in mockups creating software such as Balsamiq to get the functional feel of the application. Tests should be carried out after each main design. Designs should be improved based on the comments and feedback and tested again. Iterative designs are mostly successful.

8. Implement the design

   After the final design has passed all the tests, implementation begins. It can be broken down into smaller pieces / level / version to test and move to the next level. Step 6 should be repeated along with improvements in step 8 until desired result is obtained.

Not only design process is important but it is equally important to keep track of time the process actually takes so that we do not go over the allowed time limit. It is important to think the amount of actual time required and time given to predict and calculate / determine time required for each step of process. Gantt chart helps to keep track of what step to do when and how long it takes to accomplish each task. Figure below, Figure 3 Gantt chart, shows a Gantt chart for the design process discussed above:



Process Steps	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10
Study of design requirement	🟥	🟥								
Research and analysis on potential end users		🟦								
Finding test group represantative		🟪	🟪							
Use cases and test cases			🟨	🟨	🟨	🟨	🟨	🟨		
Mockups creation				🟩	🟩	🟩				
Implementation							🟦	🟦	🟦	🟦
Testing				⬛	⬛	⬛	⬛	⬛	⬛	⬛

Figure 3 Gantt chart

The chart above is for a small project which can be done in short period of time of about 10 weeks. The time is an estimated example time. From the chart with its distinct use of colors it is clear what step is done in what week and what tasks are ongoing simultaneously with what other tasks. It helps to keep track of steps and time and in not going over the time limit.

Note: During the implementation phase, not only user interface design is integrated to functional design but functional developments and simultaneous unit tests are also done such as database creation, authentication and authorization logic creation and other features creation and their implementation.

2.3 Cross Device User Interface

Technology has given a lot of different choices today so there is no single way of accessing data. There are also a lot of options on devices we choose to browse the information. People are using a lot of time on the internet. The availability of different devices to perform same task has increased the number even more. In general, devices people use today to interact with web are pc, mobile phones and tablets which also have different varieties within them. The range of devices users are using creates a compulsion for designers to design the interface to support cross device usage if they want more users to use their designed application. This is a service provided to the end users. There are certain challenges that come across the design process which should be faced and handled.

Cross device user interface designing is not just an approach designers follow out of compulsion. Designers want to design rich user experience and add a touch of variation to traditional approach. There are different ways to implement the cross device design. Some of them are:

Same content everywhere – The idea is to have same experience throughout all the designs and just to make the application usable in all digital devices. [7;9]

Unique experience per device – Each device can have different features and / or feel to make user experience unique across different devices. [7;9]

No matter how high the competition is, it always better to make usability successful in one device first than having support for multiple device with poor usability. Time always flies when designing and implementing. So, it is a good idea first to design and implement support for one system.

3 Elements of Interface Design

3.1 Usability

Usability is the key element of interface design. Asking simple questions can help in making the application usable. The questions could be such as is user able to navigate from one page to another, can user remember the last page he visited and so on. It deals with navigation and process of visual and functional aspects [2]. It is the quality measure attribute that measures the ease of use of user interface of an application or a web document [1].

Usability is influenced by several factors and something might go wrong. There is no guarantee that the users will stick by the designs all the time. If they do not like something they might never visit the site again. It is difficult to please them. It is better to try and assure that there are ideally no usability issues or practically minimal and unnoticeable issues. Usability can be divided into five sub quality components: 'Learnability', 'Efficiency', 'Memorability', 'Errors' and 'Satisfaction' [1]. These components describe the need of usability in interface design. They are defined below.

Learnability

Learnability is measured by the learning curve. Learnability defines how easy it is to learn a system when coming across it for the first time [1]. The basic tasks performance should be easy in a usable design. Learnability is an important factor. Quicker the user can learn, better the design is considered which is beneficial for the application or site owner. The learning curve or learnability relies on following components [25]:

- Familiarity

  It means how familiar the terms used in a system are. Use of familiar metaphors is a good example for this which has been explained in chapter 2.1 User Interface design considerations.

- Consistency
  The components in a usable system should be consistent [22,127]. Consistent design results to familiarity which makes the design usable [25]. More of consistency is described in chapter 2.1 User Interface design considerations.

- Generalizability
  It is the mixture of consistency and familiarity (not only in terms but navigation / path) in global level [25]. It is important because people can relate the new system with the old and navigate well which increases the usability of the design. This does not mean the designed system should be a copy of the existing systems or applications.

- Predictability
  Having familiarity, consistency and generalizability does not make a design usable. The system should be predictable. The users should be able to predict what a certain button will do or what a certain link will do or what a certain term means [24,15]. For example, if there is a link that says help, users expect it to have some user manual or provide some kind of help. If instead of providing help, it takes users to products or home page, it will upset them. The designer might have come up with his own meaning of help, but this should be avoided because this will scare off the users.

- Simplicity
  Simplicity is the key to all successful and usable designs. It is easy to make complex design but is difficult to use it and it is difficult to create a simple design but easy to use it. It is always better to have fewer features and usable application that loading it with hundreds of features too complex to use. To accomplish a simple design, focus on end users should be maintained. All users should be able to use the application at least at the basic level.

Efficiency

Efficiency measures the quickness of task performance after the learning phase is completed successfully for certain features [1]. Users should be able to accomplish the tasks they want to and get the result they expect to. It then shows if the design is usable or not.

Memorability

The ability of the design that can make user remember the features memorable after the first visit and first use is known as memorability [1]. The frequency of use of certain system by a user might not be high; this results into forgetting how the system is used, how to navigate to desired pages. There are some simple ways of making a user remember the system. They are listed below:

- Providing some visual identity – human mind relates to visual icons and understand easily compared to just text example if there is a wrench sign, users understand that it has something to do with their account or profile settings or other relevant settings and will follow it the designated place
- Making the system simple – people tend to forget complex systems
- Using familiar metaphors and navigation
- Consistency – see consistency and example of consistency in chapter 2.1 User Interface design consideration

Errors

Users tend to make mistakes easily but that is not important. The important part is how the errors are handled and responded. The system should address the errors and the users with a lot of patience and politeness. They tend to make mistakes because of difficult usability. If things are well predefined, people happen to make fewer mistakes. Even in case they make errors, system should gracefully handle them.

Satisfaction

Users should not drive the system, system should drive the users. If the flow of navigation is smooth, if users get results as they expect, the use of design is pleasant [1] and interesting. This means the site / system is usable. Users are satisfied with the design or in other words, the design is satisfactory.

Other than quality sub components of usability, there are other factors that affect usability of the system. They are described below:

Application/Site Navigation

It is very necessary to understand the idea of application. What are you doing and what is the expected result are two very important questions designer might come up with during application design phase. So, it is a must to have answers of these questions throughout the design process. It is very easy to get lost when doing bigger tasks. It is therefore relevant to get some techniques that help us to maintain the connectivity and flow of the work all the time. Navigation designing is a way to plan the application with sequence of actions for example if you click button1, picture1 will open. Flow chart is one of the ways to plan sitemap. Figure 4 Flowchart shows an example of a flowchart with a simple navigation of a website.

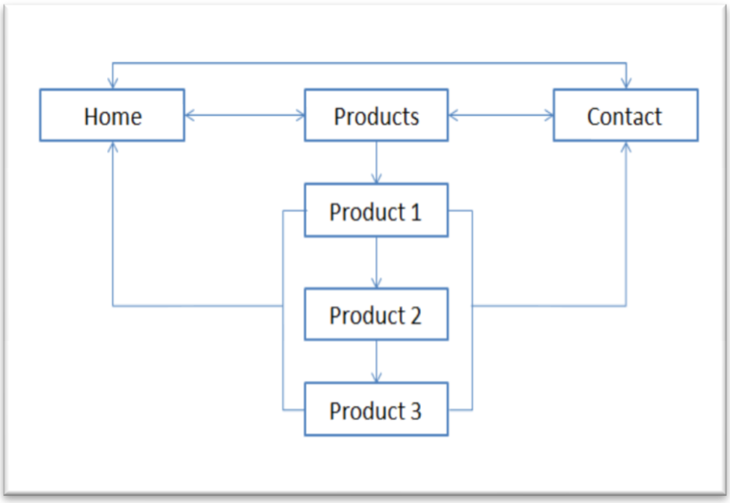


Figure 4 Flowchart

From figure above, we can get the bigger picture of the site navigation. When we are first in the home page, we have option to go to various other pages as products and contact. Products page lead us to available products of the site. The application also has a support for contacting the serving company via contact page. From the arrows, it is clear that one can get to home page from product 1 page, but cannot go to product 1 page from home page. One has to go to product 1 page via products page.

When basic navigation logic for whole application is ready, it should be tested with representative end users to see if they understand it. If they do, it means this factor will have fewer chances to cause usability issues. The system/application is usable. If not, it should be improved. It is very important to note that there should not be any dead end in navigation [32]. All pages should have options to go back to where started.

Layout Design

The other important factor to be considered for usability is layouts and how they are designed. When the idea of the application is well understood and sitemap is ready, rapid prototyping is started. The layouts give the feel of the application/site with navigation logic. Prototyping or layout designing process should always be broken down to many pieces. This gives clarity to the divided tasks. A deadline should always be followed that was assigned during UI design process in Gantt chart. The steps to be followed for a good layout design are presented below:

1. Get started with paper and pen
   - Put all the thoughts on paper

     It is very much likely to mess things up when starting the design work directly on computer with Photoshop or other software. We should not start running before we can walk, we will fall. The same logic applies here. We need to first use those resources to get started which we can throw away if we do not like them, i.e., it should not take much of our time. Paper and pen is good way. All possible ideas and thoughts should be written and sketched on the paper. Aim of paper sketching is to get clear hierarchy [27]. Brainstorming should be done to get good design work.

   - Sketch top level wireframes

     Brainstormed ideas and thoughts can be large in numbers. Some of the best ones should be chosen and top level framework should be sketched for them just giving the general idea of the application. Sketching gives clarity on system logic and on how to proceed next. Below is an example figure; Figure 5 Top level wireframe for a website:

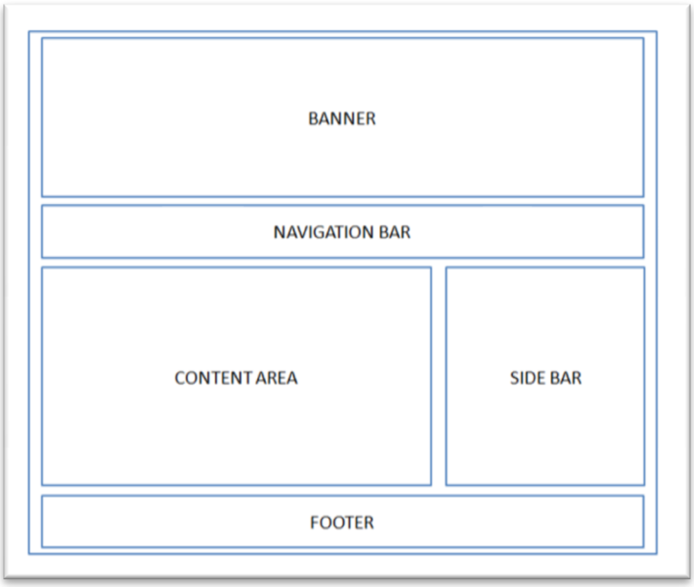


Figure 5 Top level wireframe for a website

The figure above clearly shows a basic layout of website. The website contains a banner, a navigation bar, a content area, a side bar and footer. All pages of a particular application follow this layout unless otherwise specified.

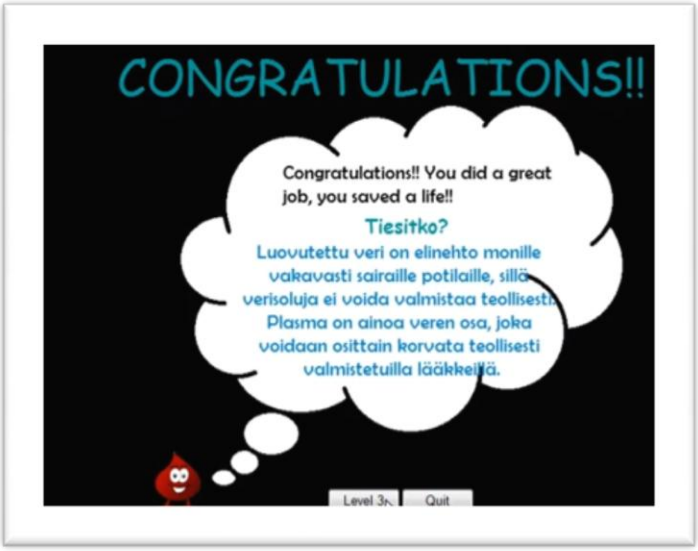
- Sketch advanced wireframes
  The top level wireframes do not include all the logic of system. Advanced sketching is required. The new sketch should contain detailed information on each page. It should give the full idea of to be created application. It should overcome all the flaws and shortcomings of basic or the top level wireframe sketching.

2. Test and evaluate with client and end users

   Sketches when completed, they should be analyzed in order to make sure there are no problems that stop designer from moving to the next level. When there is only a pair of eyes to find mistakes, it might not be enough. It is clever to go through the sketches with the client and the end users. The things that are missing, that should be omitted, comments on structure of the layout should be noted and taken care of. The problems with the design should be fixed and evaluated until desired output is obtained.

3. Continue advanced designing with computer

   The wire sketches are ready to be put on life. The common choice of software for that purpose would be Photoshop; others can be used as well. There are many free ones available. The advanced designing is about making the design as close to the real design as possible. The colors, typography and structure of all the pages should be thought of (not needed to be finalized) at this stage, if unsure, end users preferences should be followed. An example below (Figure 6 Layout design – colors, typography and structure) of one of the pages of the mobile game for children where one level has been completed:



CONGRATULATIONS!!

Congratulations!! You did a great job, you saved a life!!

Tiesitko?

Luovutettu veri on elinehto monille vakavasti sairaille potilaille, sillä verisoluja ei voida valmistaa teollisesti. Plasma on ainoa veren osa, joka voidaan osittain korvata teollisesti valmistetuilla lääkkeillä.

Figure 6 Layout design – colors, typography and structure

The figure above has some example text, example color and game’s one of the main characters. It does not need to be perfect as long as it gives the idea of the application, what it is doing at current state and how it possibly looks like. Consistent layout should be designed for all the pages of the application/web document. Layout should be designed so that it should be possible and easy to change subjective parameters like colors, typography and position later on using styling option such as CSS.

Step 2 should be repeated for advanced layout sketching until desired result is obtained.

4. Finalize the design

   Using the comments and feedbacks from potential end users and client, the layouts should be made better and made ready to be implemented (merge with functional aspect).

   Step 2 should be repeated for the final design until desired result is obtained.

   Note: the design should be tested also after the implementation to make sure everything is on place.

3.2 Visualization

There are numerous numbers of websites and web applications in use today. An individual himself uses a lot and different of them in a day. Making a workable application is not enough if one wants its application’s usability to increase. It is there where need for visualization comes in to work. Visualization is used to make the data and content of an application/system/site clear and presentable. Visualization does not just mean making an application or a site fancy but also making its content readable and data understandable easily. We can consider reading a big list of numbers; this might be sometimes annoying and sometimes difficult to understand. A simple graph might ease the difficulty. The figure below, Figure 7 Bar graph for data visualization, shows a bar graph that is used to visualize numeric data - let us suppose increase in smart phone use per year.

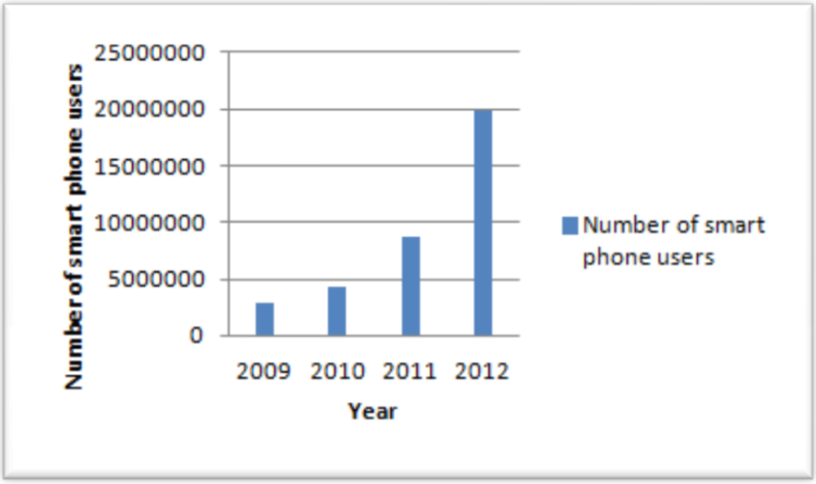


Figure 7 Bar graph for data visualization

The figure above clearly shows increase in use of smart phones has increased over time. One look at it explains the whole logic behind it while if we had to read something like below it would be difficult (note the numbers below are just an example) –
Number of smart phones used in 2009 is 2876598
Number of smart phones used in 2010 is 4360479
Number of smart phones used in 2011 is 8765420
Number of smart phones used in 2012 is 19876524

Although numbers are good for statistics it is not nice to read them in a very first glance. It makes more sense to use charts to visualize number or any other data.

Not only by bar graphs but there are several different ways for data visualization. Pie charts, spread sheets or pictures or animations. Visualization is just story telling in a better way. The very old saying – a picture is worth a thousand words – explains the strength of data visualization.

Visualization gives visual thought which plain text cannot do. Visualizing makes it easier to get the knowledge behind the object. It focuses on their meaning. It helps in seeing more, understanding more, enjoying more and stressing less. Data visualization is presentation of meaningful abstract data/information [23]. Visualization helps in communication between statistical data, users and designers. It should be used whenever possible and needed. It has pretty high importance but that does not mean over doing is good. The purpose of using visualization is to keep the content easy to understand, neat and presentable and yet simple. Simplicity is the key to successful user interface design. Over use will create a mess. Collection of end user data, their likes and dislikes and testing and evaluating designs with them here helps in determining the quantity and quality of visualization.

More of data visualization was discussed above. There is also another type of visualization; content visualization or styling. A very common way to style web documents is using CSS – cascading style sheets. Using CSS, different aspects such as margins and position, colors, font size and font style – of a web page can be manipulated to enhance the look and feel of the site. Styling as discussed above has different aspects which should be considered when trying to make the visual appearance of the design more appealing. Some of them are discussed below.

Colors

The use of colors in design works is a very subjective choice. The right choice of colors makes the end result look very elegant whereas a poor choice makes it look miserable. Potential end users should always be kept in mind while choosing the colors for the site/application. Human minds react to colors and thus colors are an effective story telling tools [5]. Depending on the end users and what the selling product is, colors can be chosen. For instance, if there is an ecommerce site that sells toys and kids wear, the theme of the site can be very colorful. Even though consistency is required, mixed colors can be used in a consistent way. Bright colors such as green, pink, blue and yellow can be used. If we are considering colors for a banking site, we cannot have a wide choice of colors. The site should have very simple and neat colors. In this case, color combinations if needed, must consist only of calm colors such as blue and white, green and white, green white and black.

The choice of colors made for a site could be changing according to seasons to make it appear thoughtful. Considering a gift ecommerce site, it can be themed white during snowy season, red during Valentine's Day, green on environment day. Colors add a bit of flavor to any plain thing. There are a lot of ways to play around it. All needed to be kept in mind for successful use is, simplicity, consistency and thoughtfulness.

Typography

Typography has almost as same effect on human perception as colors. It has power of making things strong/more important or less important. Typography includes typeface, font size, font weight, line height and spacing between letters, words and paragraphs [5]. Typefaces carry different importance. Some are designed to have high priority such as headings and some suitable for body content. Bold letters signify importance while normal text with no styling signifies normality.

Use of typography should be consistent. Overuse of different fonts on similar areas, different letters spacing will make the page lose its feel. Unnecessary use should always be avoided. CSS file should be used to manage it so that it can be easily changed anytime needed.

Structure

Structuring the pages and content boxes are as important as other styling aspects. An asymmetrical yet neat structure might represent a modern design while well and symmetric structure might be suitable for traditional users [5]. A well structured page layout can be obtained using margins and position of the content holder. A neat and simple structure is always appreciated. Users should always be considered when making unusual design choices.

Visualization is so popular and interesting things now that it is very much likely to make a lot of mistakes when applying it. Some common mistakes are mentioned below.

Common mistakes in Visualization

Over doing – Too many graphics and over use of different fonts and colors will kill the actual feel and undermine the effect of original purpose of the application/site.

Under doing it – Some designers do not put enough graphical presentation even when needed. This will lead to underrating of the built application.

Presenting without story – Designers tend to use visual effects just to make their site appear bright and fancy without having the actual need of storytelling or without actual story or data to visualize at all.

Not explaining the interactivity of visually presented data – When designing one nice site with interactive visualized content, designers forget to put information on available feature of interactivity which makes the content just static because of unknown feature.

[36]

3.3 Functionality

A system consists of visual and functional aspects. Visual aspects improve look and feel of the aspect whereas functional aspect gets the system running. We can say visual aspect is a body and functional aspect a brain. All body parts are able to perform their work because brain is giving directions on what to do and how to do. In the two earlier sections, most of visual aspects were discussed such as layout designing which is once again to provide look, feel and structure of a system. In this section, functionality will be discussed.

We can start with a question; what is functionality? When we enter our credentials on email system and click login button and it takes us to our email account; that is functionality. On entering a word on Google search field and hitting enter (clicking search), it gets us numerous data as result; that is functionality. Functionality gives life to a system. When we design layout, it itself does not do anything until it is merged with programming logic. When we are designing an interface, we are not only designing visual aspects, we are developing logic behind it too.

Adding functionality is like telling an object what task it is supposed to do. It is feature of a system's visible elements [2]. For example, if a site has help option in a top corner of the page, that option's feature is to provide help (by for example redirecting to relevant page or providing user manuals) to the users when asked for. User manual is a feature that a particular system offers by using the help element. The provided help system should be meaningful, easily accessible and usable not a compensation for a bad design [26,539]. It is easy to lose users if they do not get what they want. Providing a way to contact the person responsible for the content or design (in main level company) is a nice feature and a good way to listen to users after the application has been published. This gives users a chance to make complains about the usability issues.

The functional attributes are also used to enhance visual elements. For example, Java script can be used for data visualization. Not only does it look good and easily readable, it can have features that are interactive with the users for example having tooltips that give extra information on hovering mouse over a particular region. The figure below, Figure 8 Functionality for data visualization is an example of such a feature.

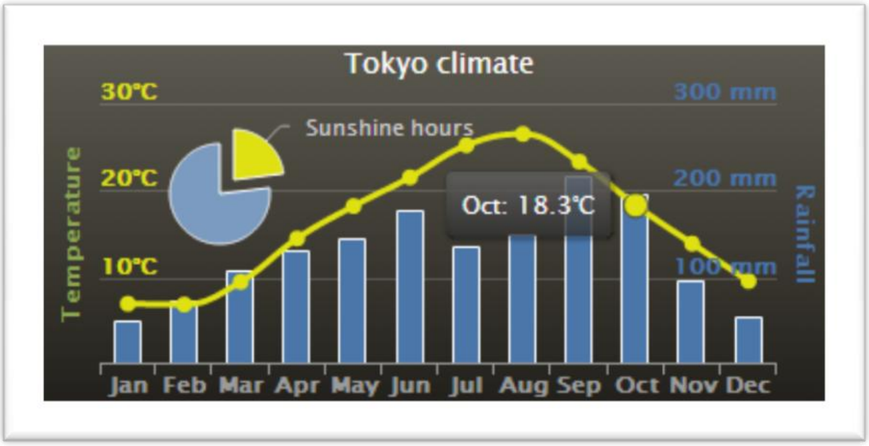


Figure 8 Functionality for data visualization [28]

The figure above showing climate of Tokyo is an example where JavaScript is used to visualize the data. The chart shows yearly amount of rainfall and sunshine in a graphical format as well has a chart inside which shows the amount of sunshine compared to rainfall. It also has an interactive feature. On pointing the mouse on each yellow dot/circle it shows average temperature for that month, in figure – text showing Oct: 18.3°C.

A common and useful functionality is search engines [2] or integrated search fields. They have search features powered by search engines or are self created. There are numerous ways of making use of functional aspect of designing but things should be always kept simple. Overdoing it brings trouble. If there is a click and navigate option on every other text of the content box, the interface will look terrible. Moreover it no longer remains usable. On the other hand, things that make using the system easier should be encouraged such as keyboard shortcuts or hotkeys. Hitting enter to submit a login form is a nice feature. It saves time and users do not need to think much to find the mouse pointer and click the go/enter option.

Responding in error situations is also a functional aspect. The system should be programmed such that it addresses users' mistakes and provides meaningful and concise error messages in politely. Rude and complex error messages will get the users mad at design and designers and force them to leave.

3.4 Accessibility

Designs are not made only for the biggest user group or users with high-tech resources. Not everyone is able to access the system for various reasons. When designing, all kinds of users must be considered and taken care of as much as possible. All those who do not have as much knowledge on computer as others should also be considered. Those with slow connections should also be considered during design work. Those with disabilities should also be considered. This needs to be done so that some kind of alternative is provided to them [2] so that specific users can still enjoy the content as much as others. There are certain ways how accessibility can be achieved. Below is a list of users for who are benefitted by accessible system:

- Visual impaired people, Hearing and speech impaired people
- People with temporary or permanent injury preventing them use of mouse
- Touch screen users
- PDA and web-enabled phones users
- People with problems like dexterity and cognitive
  [29;31]

When there is a problem there is also a solution. Below is a list of some of the possible solutions to achieve accessibility:

Keyboard navigation – Keyboard navigation should be provided to users who cannot or are less likely to use mouse. Features such as tabbed navigation [30] to navigate from one field to another, one image to another should be used. Use of hotkeys and keyboard shortcuts also help in avoiding use of mouse.

Alternative texts to all graphical objects [30] – Sometimes because of various reasons, graphical objects such as image are not loaded. People will miss out the meaning of the image on that particular space. Providing alternative texts helps users know what the image is about. It is always better to put some descriptive text than just saying it is a kid. Instead, a kid playing with colors makes more sense. It is better not to use any image links or they must be provided with alternative description text when used.

Alternative page to page with controls and applets [30] – If a web page contains a lot of controls and applets that require additional software, it is better to have an alternative page that contains plain text and simple graphics only. Not all users like to install extra software on their machines.

Alternative pages without table – Sometimes complex tables are likely to be ignored so it is safe to have alternative pages that contain no tables, just table information in plain text. This was there will be no problem if an html table is ignored. [30]

Support of reader's/user's formatting options [30] – Designers cannot rely on particular font styles or colors. Not all computer or browsing systems support all kinds of formats. For the same reason, plain texts with html heading tags are more appropriate than formatted texts. The size of application window should not be fixed. It should be relative to the user's machine.

No compulsory need of style sheets – In cases where there is no support for style sheets for example using CSS3 in Internet Explorer where all features are not being supported, it is better to confirm that the page is usable without CSS too. If not, users should have option of alternative pages without CSS. [30]

Possibility of increasing the text size and zooming options on maps and pictures – There should be option that increases the text size and zooming options should be available for maps and pictures for people who have difficulty in reading small texts or images.

Closed captions, Screen readers and Volume control – It is good to provide closed captions should be provided for important audio visual contents when possible. An alternative to closed captions could be transcripts for the audio contents. Screen readers should be implemented for those readers who are unable to read properly. An option of volume control should be available in all audio contents. It will help those who have some sort of hearing impairment. [30;31]

3.5 Interface Design Success and Failure examples and reasons behind

Interface design is a difficult job when aimed to make it simple and to get more users to use the application/site. It is easy to just put design elements together without thought and let users bear all the trouble. It is easy always easy to make bad designs. Nobody aims at making a design bad but in the process because of not considering things beforehand designs turn out being unusable. On the other hand, good designs are well planned. They are simple, well navigated and well structured. They follow all interface design considerations and take into account all the design elements thoughtfully. Below are presented some good and bad web design examples. The examples contain my personal analysis on the design.

Bad web design:
Example website: Ling’s Cars (http://www.lingscars.com/)
The website helps in contracting cars in lease in the UK. In design success or failure, the important thing that matters is the user’s first impression. As a user, when I first entered the site I was not sure where to look at. There are a lot of flashy animations all over the page and therefore it is easy to get distracted from the actual cause of being there. The first impression was bad. During the use of page, the experience sometimes gets better but this was not the case with this site. The experience only became worse. The site has every necessary element but it has been over used. Below is the list why this website design can be considered bad:

- A lot of flashy animations: This makes it difficult in focusing on what one wants. Eventually users leave the page without even trying to go to the relevant page. This also makes it difficult for users to trust that the site is genuine.
- Inconsistent: Not all the pages have footer information. The site is consistent at is using unnecessary graphics.
- Use of more than 3 font styles: This causes fonts not mixing well with each other creating an effect on user not to trust the page anymore.
- Too small fonts where needed and too large where not
- Focused on features outside main purpose: The main purpose of site is leasing cars, not allowing users play games. These side features should be minimized.
- Too much information on one page: Not everything should be put on the same page. Different tabs are created so that right things can be put under right place.

I myself would not visit the site again. Eyes start getting tired after a couple of minutes of visit. Lesser is always better unless there is need for more. Having unnecessary focus shifts and graphics will not make a site usable or even worth visiting.

Good web design:
Example website: Youtube (www.youtube.com)
The main good thing about the site is it has matched the purpose of the site. There are no unnecessary focus shifts or extra elements that repel the users. Below is the list why youtube is well designed:

- Purpose is met
- Well structured and simple
- Easy and working navigation
- Easy to become part of it (user registration, signing in and video sharing)
- Consistent design
- Clear links meaning
- Correct use of colors

Eyes do not get tired even after hours of use of the site. Clear and simple design make the experience better. It has all aspects needed for a design to be considered good.

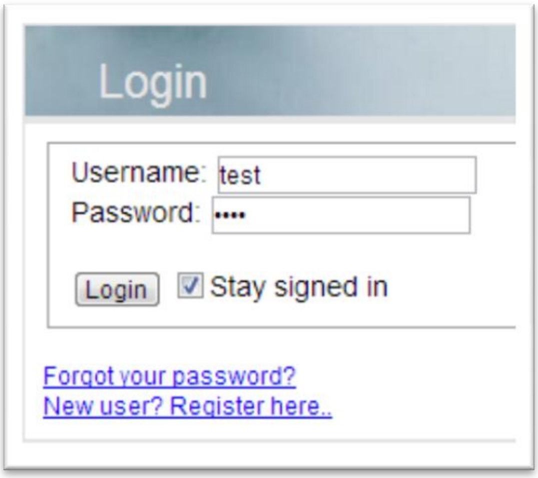
4 Access Controls and Security

4.1 Access Controls

Access control is an important design factor of an application which limits and controls who can access what resources in a certain application or a system or a geographic location [8]. Web access control restricts access to web application resources. It prevents wrong user or group from entering the application and accessing specific data. It can be categorized into two parts when talking about web applications, authentication and authorization.

Authentication

Authentication refers to identification of a certain application user to get through to the application or system. The very common way of authentication includes username and password verification, figure example (Figure 9 Authentication process) is presented below. This way right group of people are provided with right user access rights.



Username: test
Password: ••••
☑ Stay signed in

Figure 9 Authentication process

The figure above shows a login process. Username and password for a registered user are stored in the application database and they are matched when user inputs his information and clicks the ‘Login’ button. When the match is true access is allowed otherwise access is denied and user is asked to login again or application responds according to its programming.

Getting in to the system varies according to the need of the system. Usually the login process is a way to get into web systems. Taking an example of real world, getting into home needs key and keys are only provided to those who are applicable of receiving keys. They need to meet some criteria before they can get it. Similarly, in web applications, users need to meet some requirements and pass through tests to get through the authentication process. For obtaining login credentials, users need to provide their identity, detail of identity data varies according to system, during registration process. For a net banking system, bank provides user credentials only after users are verified by providing their name, home address, telephone number and social security numbers while an ecommerce site may not ask for so many details but email address to register and shipping address to ship the product where needed.

It is important to make sure no stranger gets to our homes but it is also important that our guests do not have access to our personal matters. Authorization deals with such matters. It is explained in the section below.

Authorization

Authorization refers to the right provided to specific authenticated users to use different resources. The rights can be different based on the users and user groups. An example can be two common user groups in an e-commerce site, admin and general users. An admin gets authorized access to all general users’ information and all product information of the site, can modify them, delete them or add a new one, accept or deny users requests while general users get access to only browse the products and place orders. System’s or applications’ administration decide what rights to give to what user, full or limited or none. Authorization allows specific resources to be used by specific users.

Authentication and authorization both are ways to provide security to the application system. Authentication assures that no any unverified user gets in to the system. The choice of verification is different and authorization assures no wrong user get access to resources he is not allowed to get. Other security concerns and their preventive measures are discussed in the chapter Security below.

4.2 Security

Planning, Specifying, Designing, Visual and Functional implementations all go in vain if security issues are not well handled. Web users are not always good ones. We cannot trust them all. Some may even naively input potentially dangerous threats to the application system. It is better to ensure beforehand that nothing such happens. Better safe than sorry.

There are many possible threats to our systems or applications that might ruin our businesses. Things like leaking our supposed confidential information and breaking our application data base is not pleasing news to hear. There are many such security issues that should be taken care of. The chart below, Figure 10 Application vulnerabilities, shows the application vulnerabilities population in 2013.

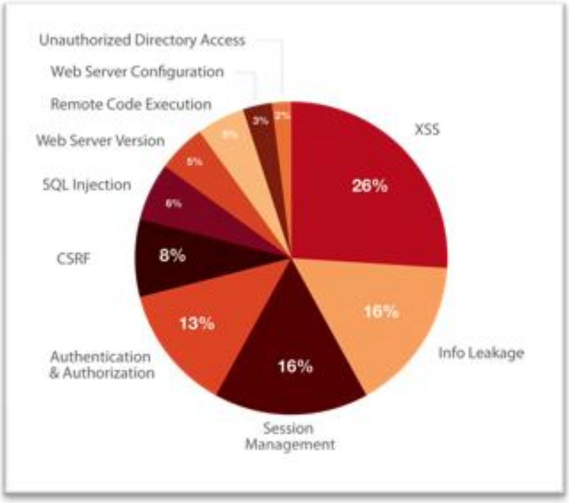


Figure 10 Application vulnerabilities [13]

From chart it is clear that cross site scripting (XSS) is the most common one followed by info leakage, session management, authentication and authorization and others. Details of the most common issues and their prevention measures are written below:

Cross site scripting – Cross site scripting is a way of executing script (usually JavaScript) and running harmful codes from external site by tricking the application users and either stealing information or pushing malicious software to user’s computer. The attackers are able to steal authentication cookies and manipulate them against users. The best way to prevent this is to filter and validate user input, allowing only safe data from users. [15]

Session Management, Authentication and Authorization – Authentication and authorization are controlled by sessions. So, it is better to explain them together. Via XSS, it is clear that attackers can steal user's authentication information and use it according to their will. Unauthenticated and unauthorized users will be able to access or manipulate information.

Attackers need to have active sessions for using their ID's. In order to prevent attackers from using the active IDs it is better to set session time out after certain period, shorter the better. Users should also be given an option of manual session ending, i.e. log out option. This will not leave user's session hanging in ready to be attacked. Even though sessions are ended, some information can still be accessed via web browser cache. To prevent browsers to cache information, the http headers should contain no-cache or no-store information. If web browser cache should be used, they should not be allowed to store session identity numbers.

SQL injection – SQL injection is a way of injecting potentially harmful SQL queries to database and damage it. Having deleted a table containing all products detail would be a nightmare but it is possible to prevent SQL injections. Even though the chart above, Figure Application vulnerabilities, showed very less percentage for SQL injection attackers still attack as implementers fail to implement the preventive measure and test it. A simple way to prevent this is validating users input before saving them to database.

5 Testing

Testing is one of the most important parts of any application design and implementation. It can be divided into two main sections: User Interface Testing and Functionality Testing. But before going through them, it is very important to know the main need for testing and what kind of tests do we need to carry out in order to assure the quality. The main aim of testing is quality assurance for a bug and usability issue free application/system. The other need of testing an application before it is in use for all is to be cost effective. The issues found in application untested in early phases are expensive to resolve than those found and resolved before publishing. A tested solution with known issues is always better than untested ones without problems. Untested solution will eventually come up with issues that were unknown earlier.

To carry out successful testing, a proper test plan is needed. Some sort of process is needed to make sure all components have been tested and tested effectively. Testing Process Design is a topic that deals with it and it is explained in chapter 5.3. For making testing easier testing tools are available for help. Testing tools are explained in chapter 5.4.

5.1 User Interface Testing

User interface testing is about testing the application interface via user's point of view. User interface testing mainly includes testing the overall appearance (look and feel) and the sequential navigational flow of the system. There is still some detailed testing that should be done to assure not broken and issue proof interface. The main aspects that should be minutely tested are navigation, browsers compatibility and layout and content [35]. They are described below:

Navigation Testing

During navigation testing, all links of the design layout must be clicked and checked that no page is orphaned (page which does not have links in other pages to be redirected to it). It is of no point to have information in a page which cannot be accessed by any links of the entire system. Thus it should be checked many times by the designer and representative users that there is no such things found. It should also be checked that there is no dead end (dead end - no option to go back to previous page or any other page). If this happens, this might irritate the users and there are chances that they will never revisit or use the system.

Navigation should be smooth and pleasant i.e. usability testing has been passed. This will assure there are less or ideally no errors. Navigation testing confirms that all links go to the right direction and navigation is usable i.e. it is predictable and obvious when clicking on something will result expected result [24,12]. This test also assures that there are no two pages displaying same content and linked differently. This is likely to happen when working for big projects without proper planning and record keeping. Important thing to note is with each test meeting with the end users, the design should be improved based on their feedback and comments. This will improve the quality of the design. In navigation testing, if all implemented keyboard shortcuts are working or not should be checked. All possible paths should be followed and tested and no flaw should just be left just like that if found. It should be fixed.

Layout and Content Testing

The other important thing to test is that designed layout fits in well with the functional logic of the system. After unit tests have been passed, integrated tests should be run to make sure all the functional logic just fits right with layout and content. In layout tests, the things needed to be tested are page structure – margins and position, color and font compatibility, color combinations, general appearance and all other similar aspects. No attribute should be left untested.

Content testing includes confirming information accuracy including spelling and grammar check [35]. The information put in the site is authentic or not should be checked from the client. Checking information accuracy is not very much designer’s job, it is client’s job but since designer puts in all the effort to create a usable site, it is his or her duty as well to confirm if the information is correct or not what client gave.

It is as important to check graphical content as textual content. Designer should make sure the images and graphics are working and there is no broken link to any image. All the images should have alternate texts. This should also be confirmed by checking and testing under different situations for example in slow and fast connection. The images used should be vector image as long as applicable; this will help prevent distortion of images on zooming in and out to very extent.

Browsers and Device Compatibility Testing

A web solution should always be working in different environments particularly in different browsers and different devices. Application should be tested in different browsers available and it should be made sure that it works in all of them. It should be tested in the browser which only supports minimum new features. In any case if the test fails, alternative pages for the web document, with minimal features in them as well, should be made available. The user should not be made disappointed but an alternative should be given.

The size of the application should be tested on different devices. The size should always be relative so that it fits in with the browser and different sized devices accordingly. If the application has been designed for mobile devices also, it should be tested in different devices commonly used: computers, tablets, pads and mobile phones and make sure that it works smoothly in all of these devices.

5.2 Functionality Testing

Functionality testing shows how well the logic of the application works. Developing an application is not enough. We need to know how well it works already before the real users start to use the application. It is more of a quality assurance process. It checks of it meets all the specified functional requirements in the Specification/Requirement document [34]. There is a list of tests to be done in order to confirm that the application works perfectly. They are Links Accessibility Testing, Forms Field Validation Testing, Cookies and Session Testing, Server Communication Testing, Security Testing, Database Testing and Performance and Stress Handling Testing [35]. They all are described below:

Links Accessibility Testing

All links available in the application should be tested thoroughly. It should be tested that there is no dead end, no orphan page and no broken links [35]. All hyperlinks, buttons, image links, map links, links within the system and all external links if used any should be tested [35]. No links should be left untested. All of them should be accessible.

Forms Field Validation and Features Testing

Forms like registration form and login form should be validated so that user does not input harmful or wrong inputs. Example email addresses during registration can be validated using regular expression. The idea is to avoid all possible troubles that can be caused by possible user error.

An application consists of a lot of features such as administration rights to modify products and user information in an ecommerce site. Other features are search, help option and many more. All of these features should be tested without failing. This will assure the application’s usability design becomes almost flawless. Features are the soul of the application. They should not be left untested or unfixed after the bug has been found. All the problems should be solved.

Cookies and Session Testing

If the application uses login or is an ecommerce application using cart systems, it uses cookies and sessions. Cookies and session management should be tested so that the users will not face any problems when logging into the system, stay logged in, buy products and not get mixed with some other users. They should be tested efficiently before the application can go to production. Even when the application is up and running, these tests should be repeated to make sure there are no problem.

Server Communication Testing

Application is uploaded to the server and published. When user performs a task on the interface, it should get response to it i.e. user interface, application server and database server are communicating. If not, there must be some connection issues which must be checked and fixed.

Security Testing

Security testing includes checking if any attempt that is harmful to the application or system is being made. If yes, they should not be allowed. We can assure a secure system by validating the user inputs, doing session managements correctly and testing all other possible security threats define in chapter 4 Security. Basically we need to try to input all possible threats / perform negative tests and see how fault tolerant has the system been designed and implemented. Implementation should be well done before starting any of such tests.

Database Testing

Database testing is testing the connection between application and database. We need to check for example all products are loaded in the product page in an ecommerce site and on checking specific product details, it is returning results. This means there is communication between the application and database. The other thing to check is if some links has been missing between tables or incorrect data has been stored in the database [35].

There should be no problem in browsing, adding them to cart or buying products. Similarly, for admin there should be no any issues while adding a new product or modifying or deleting them. If any issues occur, they should be fixed immediately. The main idea is to ensure that data integration, data update and retrieval in database via user interface, work correctly. [35;33]

Performance and Stress Handling Testing

Performance testing includes tests which checks how a system or an application performs at the time of load [35;33]. This can be tested by running the applications with different connection speeds and see if it works well on all of them. If not, modifications should be made such that it works on both high and low extreme situations usually creating alternative solution. It can also be tested using a lot of users to work on the application/site at the same time and see how results come out to be like [33].

Stress should be well handled by the application. Stress test is done to check how an application system recovers from crashes caused by the stress [33]. Stress testing is important to assure no external user intending harm to system or confidentiality of system is able to become successful in his bad intentions.

5.3 Testing Process Design

There are several ways and to test an application. The basic idea remains the same; how to test all aspects of the application. Following the simple steps can make the testing process smooth and successful. Testing process does not go one after another; there can be parallel tests ongoing along with development and other tests. Testing process design is a well documented design plan containing ways on how to proceed with testing or in other ways how to test the created solution. Main steps are described below:

Planning

Planning is a very important step in testing process. It determines how much in flow the whole process goes. Test process design with improper planning lead to failing applications. There are certain elements which should be planned before moving to actual tests; use cases, user representative, user observation, questionnaires, interviews, surveys [21]. All of them are explained below:

- Use cases
  The first thing to do before moving into tests is plan what things are needed for us to be able to complete the tests successfully. Possible scenarios should be thought over. They should be well written and documented for future use. These scenarios which explain the system's or application's behavior on certain user actions are called the use cases. Sets of such scenarios should be created beforehand.

  Based on created use cases, test cases should be created. These test cases are smaller and detailed version of use cases. Use case defines a bigger scenario while test case defines smaller and concise scenario or a use case done for a specific set of data [17]. All possible cases must be written down. It is possible to miss some cases which can be added later when found later during testing or even during end user interactions and studies.

- User observation
  It is indeed an important thing to know about the possible users of the design. It will help to make the design better with their opinions on different things, their perspective, their likes and dislikes, their way of doing things, their choice of equipments precisely electronic devices. To get the information on these matters it is sensible to find them and study them in their place of work. Their behavioral study will give hints to better system design of their choice.

  Studying set of people might be considered impolite and insecure by them when they do not know the why someone is watching them [14]. They should be informed beforehand the idea and motivation of observer's presence so that they do not feel uncomfortable doing their regular work. People and their privacy should always be respected no matter who they so any activities that might distract or disturb them should immediately be halted. It should also be decided beforehand if user study will be made by direct interaction or indirectly or in both ways. This will prepare uses for any possible conversation.

User observations should be made at the beginning of the design process. This will ensure that on results based on observations, tests can be carried out efficiently. For example, test questionnaire will contain questions that can be understood by the users as questions will be made according to their knowledge and skills. It is fairly possible to misinterpret collected information or to miss out important information despite careful observation so a guideline while planning, field study and analysis processes should be followed in order to avoid these problems [26,122-125].

- User representative
  It is not quite professional or good to test everything by the same person who has designed or developed the application. When one knows too much about expected results, one tends to miss out possible human errors which could have been made when trying out things differently. It is always a good idea to find other people who can test the solution for the designer/developer. When they know nothing about it, they are more like the users who will be using the application after it has been published.

  To be more specific, set of potential end users should be gathered. They will fairly represent the users who will use the developed product. It is also possible to include the customer or the boss himself to the group. Different people come up with different ideas. More ideas and more feedback will help making a good application. Different minds do the things differently, so there will be variety of test results. This will give the designer and developer the chance to improve the design and logic to fit in their needs.

- Questionnaires
  Questionnaires are one of the ways of getting answers to usability related questions in testing process. It contains a set of questions that asks about the user interface and application logic. The idea of asking users to answer application related questions as they go along each aspect of application is to find out how they react to the application and they find their way out in the new system.

Users might always not understand the question so they should always be assisted and helped to understand it. They should not just be handled the questionnaire and left them unattended. This might leave them feel left out and they might become unmotivated to complete the test. They should always be accompanied throughout the test, made comfortable and help should be offered whenever needed.

- Interviews
  Interviews are always a way to get more feedback from users. This can be part of user observation process or interface and logic design evaluation process. Interviews allow us to understand the feeling of users towards the system which may not have been reflected fully through the questions in the questionnaire set by designers.

  In order to have more information from users, interview questions should be prepared beforehand. Place right for interview should be arranged. The interviewees should be made comfortable. Providing beverage or snack is a polite gesture showing we care and appreciate your help for us. Always questions relevant to subject matter should be asked. Nothing personal or disturbing questions should be asked. Thanking them after the interview is over should never be forgotten.

- Surveys
  In situations when there are large group of users, there is no time for interviews or personal interaction with the users is being avoided, online surveys are a good way to get information from users. This will also give users some privacy and comfort that they are not around the designers and do not have to feel pressurized of feeling bad on telling actual facts. It is always better to get the surveys anonymously. There are many people who do not fill in survey questions because they are scared that what they thought about certain matter will be discovered later. There are many free online surveys creating applications such as Google Consumer Surveys. They also create visualized form of the results such as bar graphs. This will help to analyze the results.

Unit testing

Unit testing includes testing the features and functional logic of application usually during development process. Smallest block of functionalities and features of the application are isolated from other functionalities and features and tested separately and seen how they work individually [12]. They should be able to perform alone to pass the tests.

Interface testing

This testing includes testing the user interface of the application. It has been discussed in detail in the chapter 5.1 User Interface Testing.

Integration testing

Features and functionality of the application are combined together. All blocks tested in unit testing are put together to see if they merge well. The integration of the elements should work as well as they did alone. User Interface design is also merged with logical aspect and checked if they work. Integration testing helps confirm the working wholeness of the application.

Documenting and analyzing

It is equally important to document and analyze the result as it is to test the application's usability and study user's behavior. The documented reports help not to forget what the results were and to analyze them and use in issues solving. The documents can also be used for future referencing as a base for design improvement [10,13]. There are different ways of keeping the record of user studies and test results. User observation and interviews can be either audio recorded or audio and video recorded [14]. It is also possible to use simple paper and pen (or electronic notebooks) to take notes. The questionnaires and surveys can be saved as they are received.

The saved documents should be analyzed in order to get to next step of design and logic improvement, design refinement [22,339] and noticed problems solving. Enough time should be allocated for this purpose. No conclusions can be fully meaningful which is extracted out in a rush. If there is some problem in understanding user's point of view on something, it is better to discuss with someone else too without breaching confidentiality of content.

Issues Fixing

Issues fixing is not directly part of testing process but is indirectly related to it. When tests are done, results might have some bugs or implementation issues which should be fixed in order to get the application fault proof. After the fix, tests are iterated. This kind of forms a connected cycle. Issues fixing is thus considered as an indirectly connected part of test process as without it, testing cannot get the right pace.

Regression Testing

When issues are resolved, the tests to check the fix should be implemented again. This is regression testing. Regression tests should be quick and for the same test case used to find the problem; this will help in keeping the testing process moving. During the tests it should also be checked that no other bugs or problems are found. If there are problems, they should be found and fix should be tested again until issue free results are obtained. It is not good to leave a known problem making a faulty or a buggy system as systems already have unknown problems that might trouble in the future.

The testing process is also shown in a visual form below; Figure 11 Testing Process:

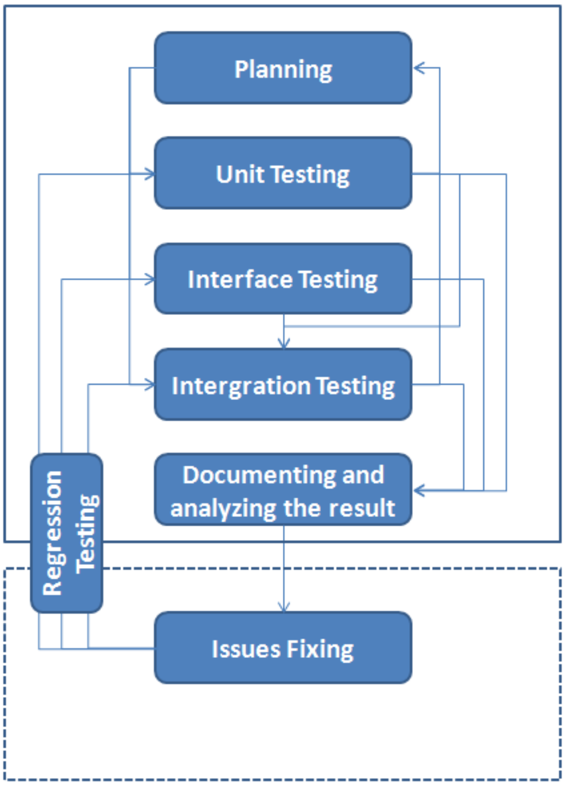


Figure 11 Testing Process

The above figure clearly shows connection between each testing step. The arrows show the inter-connectivity between them. It shows how most of the steps are not just occurring after completion of one but are occurring simultaneously or in a parallel manner. For example, unit testing and interface testing can be done at the same time. They do not need to be waiting for another to finish first. On contrary, in order to carry out integration tests, part of both unit tested result and interface tested results are needed.

It is also possible to skip out unit and interface testing and directly jump into integration tests but this will end up in pretty complex test scenario as all smaller tests that have been done on other two should be repeated here. The chances are testing will never get done or full proof results are hard to obtain.

5.4 Testing Tools

Testing is a complicated, stressing and time consuming process. It is easier to test a web application for different components using some automated tools that will make things simpler. There are different kinds of options available. They are also classified into different categories according to test needs. Some of important tools with examples of available tools in market are as follow:

- Load and Performance Test Tools – This tool tests load handling and performance of the application. Example tools in market: Telerik Test Studio, HttpRider, Loadtracer, AppViewWeb
- Link Checkers – This tool checks that there are no orphan pages in the application. Example tools in market: W3C Link Checker [3], LinkScan
- HTML Validators – It validates html tags and CSS. Example tools in market: W3C Markup validator [3], Site Check, Html Validator, WDG HTML Validator
- Cross-browser Testing Tools – It checks browser compatibility. Example tools in market: CrossBrowserTesting, TestingBot
- Web Functional/Regression Test Tools – It helps in functional aspects testing and in regression testing as suggested by its name. Example tools in market: Jubala, FuncUnit
- Web Site Security Test Tools – It tests security issues. Example tools in market: Vega, Seeker

[11]

6 Application Design and Implementation

Enoro (the company for which the thesis work is done) supports an automatically scheduled system monitoring job which notifies users via email about all the failed and successful jobs running on a server. The application generates a log file and a monthly statistics report. The only way to get the information outside of the application installed system is via email. The System Monitoring Application does not have a web interface.

The idea for this thesis came up when web interface for the application was to be designed so that users can easily log onto their respective systems and keep track of all their servers' jobs statuses without having to browse through all the emails in the past. The designed interface also allows viewing results of different servers under one common platform. The current system is missing this feature and thus further design and development was needed. Detailed information on features of the application is in application development process. The application development process consisted of Requirement definition, Design and Implementation phases and Testing.

6.1 Requirement

System Monitoring Web Application's main requirement is – "Users must be able to see Monitoring result from a website". The main purpose is to enable users to keep track of monitoring result via web interface. The requirement must be accomplished fulfilling certain criteria. They are listed below:

- The application should be based on Enoro's existing System Monitoring application which is installed to client server.
- Comma Separated Value (CSV) interface should be defined and designed for current application output (current application supports a log file overwritten per run and monthly statistics XML file)
- The System Monitoring application for web should primarily run on Enoro's Generis system which later will be replicated to web.
- There should be support for importing defined CSV data model
- The data model should be imported and saved using Generis Time Series Products and Time Series Product Instances via Generis Functions and Advanced Search queries
- User interface should be designed using Enoro Control Display
- The created Control Display should be replicated to web

6.2 Design

Designing is a very crucial part of an application development process. If the design is right, implementation becomes smooth and application usable. Based on requirement, the application design was divided into five main parts.

Design of the System Monitoring Application changes (CSV file output from existing application)

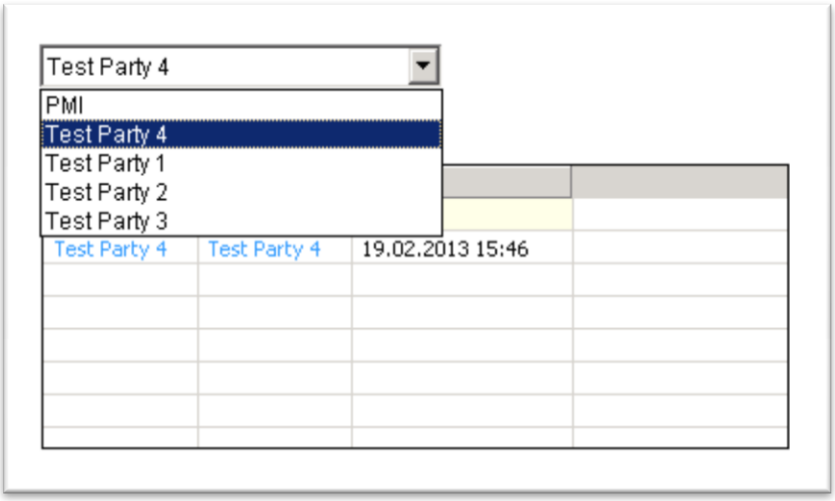
Existing system monitoring application has output support only of an overwritten log report file per run and monthly statistics XML file. The need is to have a CSV file per scheduled run of the application. A decision to use CSV file was made because it is easy to read by Generis and to save the file data to the system. What information from system monitoring application is to be extracted to new CSV file was also decided. The CSV file format is designed which will be used to import monitoring result to a common Generis system regardless of customer system.

Design of the CSV files import, Time Series Products and Instances Creation

After the SM application exports desired CSV output file, the next step is to import it to the Generis system. It is done by Generis functions. SM application creates the output file on a certain output folder. The file is put to a certain input folder either manually or automatically via FTP transfer. Allocated Generis IFM job detects the file on the input folder and executes the function responsible for reading and saving the file data to database. The function then reads the file, verifies the information, creates time series products (TSP) and instances (TSPI) and saves the file information to them. The TSP and TSPI are designed based on the data model.

Design of the Control Display layout for the data model

The result from System Monitoring is stored to the database after implementation of previous step. But there is no user interface to access it. Generis Control Display is used to create user interface for SM result. A simple Control Display is created with combo box and list controls as CD Web replication API only supports these objects. Web replication API replicates CD created in Generis to web readable form. An example of combo box and list control can be seen in the figure below; Figure 12 Combo Box and List Control.



Test Party 4 Test Party 4 19.02.2013 15:46

Figure 12 Combo Box and List Control

Design of the Control Display web replication for the Generis Control Display layout

The idea of Control Display Web Replication is to run a light weight user interface and from devices which do not have Generis installed in them or, in general running it on web so that it does not require specific platform such as Windows OS to view it. This feature is provided by Control Display Web Replication Application which works in the following way:

1) The web application sends a request to the WCF (Windows Communication Foundation) service and asks for available Control Displays

2) The WCF service queries the database and returns a list of available Control Displays

3) The web application calls the Generis Control Display Application

4) The WCF service returns a list of components from Control Display to the web application.

5) The web application draws the components on the web page.

6) The user selects something in the CD and the web application sends a request to the WCF service about it

7) The WCF service updates the CD and starts again from point 4

Design of Web application user interface

The web application should contain following features (main and minor):
User groups – Database implementation -> database model design
System Login – authentication and authorization
User registration
Password change and retrieval
Feedback and problem reporting
Administrative rights for user data modification
Usable navigation
Visual aspect – Structure and graphical user interface design by CSS implementation

These five main steps of application design can be visualized through system architecture. The figure below, Figure 13 application architecture below, shows the whole process of application design and implementation.

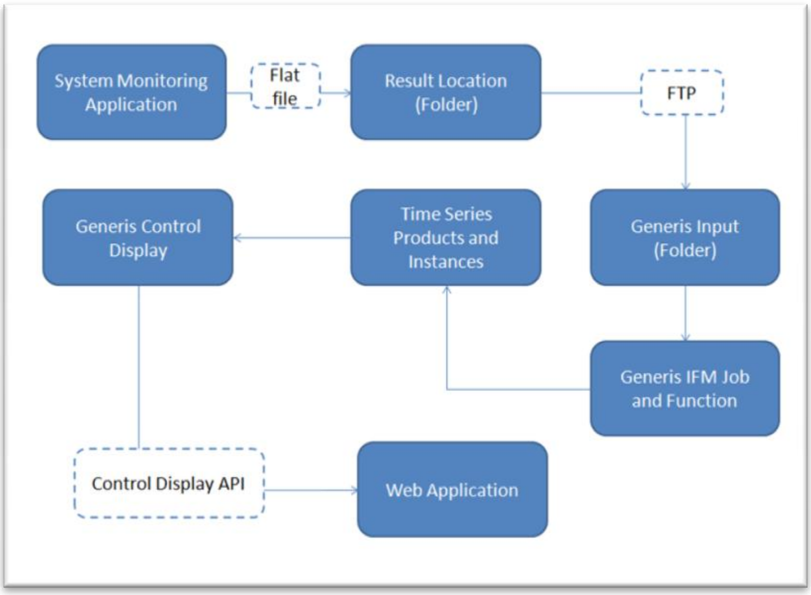


Figure 13 Application Architecture

The figure above has steps of System Monitoring Web Application development. SM application sends CSV file output to client server location which then either manually or automatically by FTP transfer is put to common server location. Generis IFM jobs and function read and import the file and data to Generis which are saved to Generis time series products and instances. Generis Control Display interface is designed and developed which by CD web replication is sent to Web. The web application consists of usable user interface consisting of features defined in the design phase.

6.3 Implementation

The design work is implemented for each design part. The implementation work for each part is explained below:

Implementation of the System Monitoring Application changes (CSV file output from existing application)

Detail of SM application changes has been explained in Chapter 6.2 Design. All the design planned for SM application changes to meet the requirement should be implemented. The codes for system monitoring application should be altered such that it produces the output in CSV format. The CSV format has already been specified in the design phase. This feature was designed in this thesis work but was implemented by an outside developer.

Implementation of the data model of System Monitoring data (CSV file import, Time Series Products and Instances Creation)

To check presence of CSV file in the designated input folder, Generis IFM job is created. The job is linked to file import function. The function reads the file and imports it to the system. The function creates respective TSP and TSPI. TSP object model is created. Without the TSP object model, TSP and TSPI cannot be created where respective CSV file imported data is stored.

Implementation of the Control Display layout for the data model

User interface is needed to view the stored data. Control Displays are created for that. The created control displays show result for each customer.

Implementation of the Control Display web replication for the Generis Control Display layout

Application for replicating the Control Display web replication is made. This has already been implemented before hand by developers. CD web replication includes only testing in this thesis part.

Implementation of Web application user interface

Every aspects mentioned in the design plan is implemented. Designed database model is implemented and database tables are created. Other features such as system login and different user rights are implemented. Users are able to register to the system, provide feedback and change their passwords when needed. Admin is able to modify user data. This implementation is done using C# programming language in ASP .NET framework. Some styling is given to the application using Cascading Style Sheets.

6.4 Testing

For testing, test cases are written based on requirement and application design. The test cases are then executed in units and in integrated form. After that a regression testing is performed. The test cases are written below for each designed and implemented steps.

Testing of the System Monitoring Application changes (CSV file output from existing application)

To make a full proof design, use cases and test cases should be thought of beforehand. Some basic test cases for full accomplishment of designing the SM application changes are listed below.

Test cases:

- System Monitoring application exports correct CSV file
- the file contains correct results
- the application is stable i.e., it does not crash on generating the report in new format
- good performance i.e., the report generation does not take more than reasonable time

Testing of the data model of System Monitoring data (CSV file import, Time Series Products and Instances Creation)

This step should also be tested in order to verify the functionality. Basic test cases are listed under Test cases.

Test cases:

- the function imports the file correctly
- time series products and instances are created
- the function saves the SM result to system as specified
- negative tests for quality assurance i.e., on purpose inputting wrong file and making sure system gives error

Testing of the Control Display layout for the data model

Listed below are Control Display test cases.

Test cases:

- UI testing
    - navigation
- Functionality testing
    - right results are displayed for right system

Testing of the Control Display web replication for the Generis Control Display layout

Control Display Web Replication should also be tested so that the final working Control Display from Generis works in web as well. Basic test cases are listed below.

Test cases:

- CD Web replication API testing i.e., making sure that the CD web replication API works by creating a simple CD in Generis using only list control and combo box, no error should be generated
- no error when using labels in Generis CD – note: replicated Control Displays do not support labels; they are rendered as invisible to cause no error
- cross browser tests

Testing of Web application user interface

Web application user interface should also be tested for all the features that were designed in design phase and implemented in implementation phase. Basic application tests that should be carried out have been presented in chapter 5 sub chapters 5.1 User Interface Testing and 5.2 Functionality Testing. Test cases listed below are extra tests performed for this particular application. Tests for 5.1 and 5.2 should be performed along with the below listed cases.

Test cases:

- Authentication and authorization
    - logging in with wrong credentials – error should be reported and person should not be able to log in to the system
    - user is able to remain logged in to the system until manually logged out
    - no one user is allowed to see other user’s SM results
    - after the user has been idle for long time, system logs the user out automatically
- User registration, password change and retrieval, feedback and problem reporting
    - user is able to register to the system
        - sending email to admin
        - no full rights until admin has assigned the user to a certain group
    - user is be able to change his password after logging into the system
    - user is be able to ask for password change before logging into the system provided that his email address is given for sending reset information
    - user is able to send feedbacks and problem reports to admin via a contact form
- Administrative rights for user data modification
    - admin can add, remove or modify users and users right
    - admin has access to all the customer’s system monitoring results

The final part is to test all of these parts combined. Overall usability of the application should be tested and the tests should be passed.

6.5 Project Analysis

The overall application development process was a good experience. The application had some good results and also some short comings. Some improvisations are required in the future versions. The project was not the final product (final product was not aimed for thesis) and has been accepted by the supervisor at work as well as the supervisor for the thesis. General analysis of project/application is below:

What went right?
Below is the list of things that went right during the application development process:

- The application designing met the requirement
- The implementation of design went as planned

What went wrong?
Below is a list of things that went wrong during the application development process:

- Not understanding the idea of to be developed application in the beginning led to extra time consumption with less productivity in the beginning

What aspects to be improved in future versions?

- Design changes should be made to meet the final product
- Styling should be improved as per need
- More functionality should be added to meet the final product requirement
- More and detailed test cases should be added and tested. Also regression tests should be done to the existing ones

Valuable learning experience was gained through this project. This will be used in design and implementation for future versions of the application and to accomplish a successful and usable product.

7 Conclusion

The project work was done to see how user interface design and implementation process can be benifitted to create a usable design. The other motive was to see how well web and Enoro’s Generis system interact. For core application, Generis system was used. For other web aspects, ASP.NET framework was used with C# programming language and web application for System Monitoring was developed.

The use of Generis system for main application made it easier to build the application with its inbuilt functions. The only problem which occurred due to use of Generis system was that more time was consumed to understand what the functions do and how are they used. The same thing happened with Generis internal jobs and Control Displays. Generis Control Display was used for creating interface for the monitoring results. The process became smooth after understanding the Generis system and its elements. During the application development, use of ASP.NET framework was also studied and so was programming with C# in the framework. Help for using Generis elements was provided by company personnel and Enoro wiki. For ASP.NET, Microsoft MSDN library was also quite helpful along with other online resources.

The final product for the thesis was not the final product of the Enoro application. The other versions will be done in the future and they are ongoing at the moment. The thesis provides the users with the idea of how the final application and functionality will be like. The application possesses business value for the company.

Using the interface design elements helped much in accomplishing the goals decided for the project. It broke down tasks in small pieces making it easier to read and do. The overall project was successful. Designing interface is not an easy job but not as difficult when right processes and methods are used.

References

1. Nielsen J. Usability 101: Introduction to Usability [online]. Warm Springs Blvd. Fremont, CA: Nielsen Norman Group; January 2012.
   URL: http://www.nngroup.com/articles/usability-101-introduction-to-usability. Accessed 7 April 2013.

2. Learning Technologies at Virginia Tech. The Elements of Interface Design [online]. Blacksburg, VA: designs shop lessons in effective teaching; April 2011. URL: http://www.edtech.vt.edu/edtech/id/interface/index.html. Accessed 7 April 2013.

3. Parr B. 7 Useful Tools for Web Development Testing [online]. NY: Mashable; March 2009.
   URL: http://mashable.com/2009/03/21/tools-web-testing. Accessed 7 April 2013.

4. Young S. User Interface Design With Markup [online]. Cambridge, UK: Machine Intelligence Laboratory Cambridge University Department of Engineering; January 2009.
   URL: http://mi.eng.cam.ac.uk/~sjy/3F6/web9.pdf. Accessed 7 April 2013.

5. MSDN Microsoft. Chapter 4: Design and Layout [online]. Redmond, WA: MSDN; 2013.
   URL: http://msdn.microsoft.com/en-us/library/hh404087.aspx. Accessed 7 April 2013.

6. Farley J. The Web Design Process Part 3: Layout [online]. Melbourne, Australia: Sitepoint; November 2010.
   URL: http://www.sitepoint.com/the-web-design-process-part-3-layout/. Accessed 7 April 2013.

7. Misener D. Cross-device strategies put all of the puzzle pieces together [online]. Ontario, Canada: The Globe and mail; September 2012.
   URL: http://m.theglobeandmail.com/report-on-business/small-business/sb-digital/web-strategy/cross-device-strategies-put-all-of-the-puzzle-pieces-together/article4102506/?service=mobile. Accessed 7 April 2013.

8. Windows Dev Center Microsoft. Access Control [online]. Redmond, WA: Windows Dev Center; 2013.
   URL: http://msdn.microsoft.com/en-us/library/windows/desktop/aa374860(v=vs.85).aspx. Accessed 7 April 2013.

9. Smus B. A non-responsive approach to building cross-device webapps [online]. USA: HTML5 Rocks; April 2012.
   URL: http://www.html5rocks.com/en/mobile/cross-device/. Accessed 7 April 2013.

10. Lawrence D, Tavakol S. Balanced Website Design: Optimising Aesthetics, Usability and Purpose. London: Springer; 2007.

11. Hower R. Web Site Test Tools and Site Management Tools [online]. SoftwareQATest.com; January 2013.
URL: http://www.softwareqatest.com/qatweb1.html. Accessed 7 April 2013.

12. MSDN Microsoft. Unit Testing [online]. Redmond, WA: MSDN; 2013.
URL: http://msdn.microsoft.com/en-us/library/aa292197(v=vs.71).aspx. Accessed 7 April 2013.

13. Cenzic, Inc. Cenzic Application Vulnerability Trends Report 2013 Now Available [online]. Campbell, CA: Cenzic, Inc; March 2013.
URL: http://blog.cenzic.com/2013/03/application-vulnerability-trends-report/. Accessed 12 April 2013.

14. Bevan N. User observation/field studies [online]. Usability Net; 2006.
URL: http://www.usabilitynet.org/tools/userobservation.htm. Accessed 7 April 2013.

15. Mark. Most common website security issues and prevention [online]. UK: Wubble You; June 2011.
URL: http://www.wubbleyou.co.uk/blog/articles/
most-common-website-security-issues-and-prevention. Accessed 8 April 2013.

16. Cox P. Maintaining Consistency in Your UI Design [online]. Design Shack; August 2012.
URL: http://designshack.net/articles/graphics/
maintaining-consistency-in-your-ui-design/. Accessed 7 April 2013.

17. Hansen J. Test Cases Based on Use Cases [online]. Denmark: All About Requirements; October 2011.
URL: http://www.allaboutrequirements.com/2011/10/
test-cases-based-on-use-cases.html. Accessed 7 April 2013.

18. Gayle M. 10 Important UI Design Considerations for Web Apps [online]. Six Revisions; May 2011.
URL: http://sixrevisions.com/user-interface/
10-important-ui-design-considerations-for-web-apps/. Accessed 7 April 2013.

19. Pratt J. Making Choices in UI Design: You Can't Please Everyone [online]. Chicago, IL: Mobile Evolution; July 2011.
URL: http://creativealgorithms.com/blog/content/
ui-design-you-cant-please-everyone. Accessed 7 April 2013.

20. Shankar J. User Interface Design [online]. San Francisco, CA: Slide Share; August 2012
URL: http://www.slideshare.net/jaishas/
user-interface-designsommerville-bangalore-university. Accessed 7 April 2013.

21. Bevan N. Methods Table [online]. Usability Net; 2006.
URL: http://www.usabilitynet.org/tools/methods.htm. Accessed 7 April 2013.

22. Mayhew DJ. The Usability Engineering Lifecycle : Practitioner's Handbook for User Interface Design. San Francisco, CA: Morgan Kauffmann Publishers; 1999.

23. Few S. Data Visualization for Human Perception [online]. Aarhus, Denmark : Interaction Design Foundation; 2013.
URL: http://www.interaction-design.org/encyclopedia/
data_visualization_for_human_perception.html. Accessed 7 April 2013.

24. Krug S. Do not Make Me Think! : A Common Sense Approach to Web Usability. Indianapolis, IN: New Riders; 2000.

25. Louis T. Usability 101: Learnability [online]. NY: Tnl.net; 2011.
URL: http://www.tnl.net/blog/2003/06/17/
usability-101-learnability/#sthash.sQoYbBZK.dpbs. Accessed 7 April 2013.

26. Shneiderman B, Plaisant C. Designing the User Interface : Strategies for Effective Human-Computer Interaction. 4th ed. Boston, MA: Pearson Addison Wesley; 2005.

27. Guglieri C. Pro tips: 20 steps to the perfect website layout [online]. England, UK: Creative Bloq; August 2012.
URL: http://www.creativebloq.com/web-design/
steps-perfect-website-layout-812625. Accessed 7 April 2013.

28. HighChartsJS. Vik I Sogn, High Charts JS [online]. Norway: Highsoft Solutions AS; 2013.
URL: http://www.highcharts.com/. Accessed 7 April 2013.

29. Erry C, Hardy B. Web Accessibility and Online Banking [online]. Sydney, Australia: PTG Global; August 2005.
URL: http://www.ptg-global.com/PDFArticles/
Accessibility%20and%20user%20interface%20design%20v1.0.pdf. Accessed 7 April 2013.

30. MSDN Microsoft. Accessibility Design Guidelines for the Web [online]. Redmond, WA: MSDN; 2013.
URL: http://msdn.microsoft.com/en-us/library/aa291312(v=vs.71).aspx. Accessed 7 April 2013.

31. Microsoft Accessibility. Accessible Technology in Computing—Examining Awareness, Use, and Future Potential [online]. Redmond, WA: Microsoft Corporation; 2004.
URL: http://www.microsoft.com/enable/research/phase2.aspx. Accessed 7 April 2013.

32. Lynch PJ, Horton S. Web Style Guide: Basic Design Principles for Creating Web Sites - Navigation and Way Finding [online]. 3rd ed. Kendalville, IN: Courier Kendallville; January 2009.
URL: http://webstyleguide.com/wsg3/4-interface-design/2-navigation.html. Accessed 7 April 2013.

33. Software Testing Social Network. Basic Guidelines for Website Testing [online]. Reston, VA: Software Testing Social Network; November 2011.
URL: http://softwareqatesting.org/web-testing/
web-testing-where-to-start.html. Accessed 7 April 2013.

34. NResult. Functionality Testing [online]. Vancouver, WA: NResult; 2011. URL: http://nresult.com/quality-assurance/functionality-testing/. Accessed 7 April 2013.

35. Software Testing Social Network. Basic Guidelines for Website Testing [online]. Reston, VA: Software Testing Social Network; November 2011. URL: http://softwareqatesting.org/web-testing/basic-guidelines-for-website-testing.html. Accessed 7 April 2013.

36. Perez R. Four Easy Visualization Mistakes to Avoid [online]. Visual.ly; May 2012. URL: http://blog.visual.ly/data-visualization-mistakes-to-avoid/. Accessed 7 April 2013.