FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Specifying an Information System for Managing Distributed Generation Equipment and Related Services – a Case Study with EDF Energy in the UK

André Moreira Reina

FINAL VERSION

Report of Project/Dissertation
Master in Informatics and Computing Engineering

Supervisor: João Bernardo de Sena Esteves Falcão e Cunha (Associated Professor)

1 August 2009

Specifying an Information System for Managing Distributed Generation Equipment and Related Services – a Case Study with EDF Energy in the UK

André Moreira Reina

Report of Project/Dissertation
Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: António Augusto de Sousa (Associated Professor)

____

External Examiner: Luis Manuel Borges Gouveia (Associated Professor)

Internal Examiner: João Bernardo de Sena Esteves Falcão e Cunha (Associated Professor)

31st of July 2009

Abstract

Nowadays, the modern organizations need to adapt their information and management systems with more frequency to increase operational efficiency and better answer to the market needs and, with this, maintain the competitiveness.

EDF Energy Networks has decided to improve and centralize its Information System regarding the Distributed Generation Energy previously done with excel sheets, word documents, small databases and sheets of paper to a centralized and accessible from the Intranet for those who have the rights.

Distributed Generation of Energy is a system with its own generators and its owners that generates unities of energy that are directly connected to the public system of energy distribution.

To achieve the success of this project were raised software requirements and the process inside the company regarding Distributed Generation Energy. Were also identified the target users of this Information System and, from them, the requirements of the project were build. Is also shown in this report the results of the planning and design together with the evolution and discussion with the target users of this Information System.

The result of this project was the specification of an Information System completely functional with a graphic interface designed to get advantages from the good habits and expectations from the users. Generally the achieved system answers to all the objectives initially established.

Resumo

Hoje em dia as organizações modernas necessitam de adaptar os seus sistemas de informação e gestão com cada vez mais frequência de modo a aumentar a sua eficiência operacional e melhor responder às exigências do mercado e assim, se manterem competitivas.

A EDF Energy Networks decidiu melhorar e centralizar o seu sistema de informação relativamente à geração de energia distribuída, antigamente organizado em folhas de Excel, folhas de Word, pequenas bases de dados e folhas de arquivo, para uma base de dados central e acessível na Intranet por todas as pessoas de direito dentro da empresa.

A geração distribuída de energia é todo um sistema com geradores próprios e seus respectivos proprietários que produzem unidades de energia que são directamente ligadas ao sistema público de distribuição de energia.

Para a realização deste projecto foram levantados os requisitos de software assim como o respectivo processo dentro da empresa relativo à distribuição de energia distribuída.

Foram também identificadas as categorias dos utilizadores alvos deste sistema de informação, a partir das quais os requisitos concretos do projecto foram estabelecidos. São também apresentados os resultados do planeamento, desenho e construção do sistema de informação assim como a sua evolução e discussão com os utilizadores alvo.

O resultado deste projecto foi a especificação de um sistema de informação completamente funcional com um interface gráfico desenhado de modo a tirar vantagem dos bons hábitos e expectativas dos utilizadores. No geral o sistema produzido responde a todos os objectivos inicialmente estabelecidos.

Acknowledgements

"The difference between a successful person and others is not a lack of strength, not a lack of knowledge, but rather a lack in will."

-Vince Lombardi

First and foremost I would like to express my gratitude to everyone at EDF Energy Networks for making this traineeship possible. In particular, I would like to thank Dr. Cristiano Marantes who opened the door and gave me the opportunity to undertake this project and for all the support he has given me. My thanks are also extended to Bill d'Albertanson for his guidance and to Awais Lodhi for the project support he has offered and for his assistance in helping me arranging accommodation when I first arrived at Bury St Edmunds.

Special thanks go to Ken Cumby for helping me with my written English vocabulary and grammar. Also for his warm welcome when I arrived at EDF Energy Networks, which has continued throughout my entire stay.

I would like to thanks everyone at FEUP for providing me with an excellent Masters Degree course, particularly MIEIC's director Prof. António Augusto de Sousa. Also special thanks are given to my supervisor Prof. João Falcão e Cunha for his support and to Prof. Raul Vidal, former MIEIC director, for his direction and welcome during my first three years of study.

I am also grateful to the people who were part of my ERASMUS experience in Rome, just before I came to EDF Energy, particularly to Gonçalo Queirós who was there for me all the time.

My gratitude also goes to the endless list of people who would make this report never ending if I actually named them all.

Last, but not least, I would like to give a heartfelt thanks to my family for their invaluable support. This is especially extended to my parents, Manuel and Manuela Reina, for not being mad at me for only spending three days back at home between my six months stay in Rome and the six months I have spent in the UK at EDF Energy Networks.

- André Reina

Contents

1. **Introduction**..........8
   - 1.1 EDF Group..........10
     - 1.1.1 EDF Energy Networks..........10
   - 1.2 Distributed Generation..........10
   - 1.3 Dissertation Overview..........11
   - 1.4 Problem Description..........12
2. **Approach for specifying DGIS**..........13
   - 2.1 Software Requirement Specification..........14
   - 2.2 Human Computer Interaction..........16
     - 2.2.1 Overview..........16
     - 2.2.2 Nielsen Heuristics..........16
     - 2.2.3 Three Principles of Shneiderman..........17
     - 2.2.4 George Miller Theory..........18
     - 2.2.5 Evaluate and test the interface..........18
   - 2.3 Platform..........19
   - 2.4 Project Priorities and Plans..........19
     - 2.4.1 Software Development Methodology..........20
3. **Proposed IS for DG**..........21
   - 3.1 Requirements – an Overview..........22
   - 3.2 User Interface..........23
4. **Evaluation of Proposed IS**..........25
   - 4.1 Requirements..........26
   - 4.2 User Interface..........26
     - 4.2.1 Nielson Heuristics..........32
     - 4.2.2 Three Principles of Shneiderman..........32
     - 4.2.3 George Miller Theory..........33
5. **Conclusion and Future Work**..........33
   - 5.1 Conclusion..........34
   - 5.2 Future Work..........35

List of Figures

Image 1 - Distributed Generation Process 11
Image 2 - Companies managing electrical distribution on UK 12
Image 3 - Rapid Application Development 20
Image 4 - Example of excel sheet page 23
Image 5 - Access Prototype Screenshot 24
Image 6- Initial page of the Proposed IS 27
Image 7 - New G59 Application Interface 28
Image 8 - New G59 Application Customer Section 29
Image 9 - New G59 Application Site Section 30
Image 10 - View/Edit G59 Application Customer Edition 30
Image 11 - Project Management Interface 31
Image 12 - G59 Application Successfully Submission 31

List of Tables

Table 1 - Use Case Structure 15

Abbreviations

DG	Distributed Generation
DGIS	Distributed Generation Information System
ERASMUS	European Region Action Scheme for the Mobility of University Students
FEUP	Faculty of Engineering of the University of Porto
IEEE	Institute of Electrical and Electronics Engineers
IS	Information System
J2EE	Java Platform, Enterprise Edition
MIEIC	Master in Informatics and Computing Engineering
MS	Microsoft
PC	Personal Computer
RAD	Rapid Application Development
SiFEUP	System Information of Faculty of Engineering of University of Porto
SRS	Software Requirements Specification
UK	United Kingdom

Chapter 1

Introduction

This chapter introduces the project with a brief description of the company where this internship took place, the project which the internship concerned and this report’s contents.

1.1 EDF Group

Born in 1946 in France known as “Electricité de France”, the EDF group is a leading player in the European industry. With a net income of 4.3 billion Euros in 2008, EDF group, present in all areas of the electricity value chain, from generation to trading, and increasingly active in the gas chain in Europe. Leader in the French electricity market, the group also has solid positions in the United Kingdom, Germany and Italy.

In the electricity sector, it has the premier generation fleet and customer portfolio in Europe and operates in strategically targeted areas in the rest of the world. The group is also the leading network operator in Europe, giving a sound business model, equally balanced between regulated activities and those open to competition. [EDFG09]

1.1.1 EDF Energy Networks

EDF Energy Networks is one of the UK’s largest energy companies and a wholly-owned subsidiary of the EDF Group, one of Europe's largest energy groups. It generates around 6% of the UK's electricity, employ nearly 12,000 people, and deliver electricity to 7.9 million customer homes and businesses through the public networks. [EDFN09]

1.2 Distributed Generation

Distributed generation also known as embedded generation is a system that involves small amounts of generation located on a utility's distribution system for the purpose of meeting local (substation level) peak loads and/or displacing the need to build additional (or upgrade) local distribution lines.

The generation of electricity is made by small-scale power plants located near the electric loads they serve. The term generally is used to refer to power plants that are small enough to be connected to distribution instead of transmission. Depending on the size of nearby loads and the capacity of the distribution line to which it is connected, the maximum size of distributed generation can vary from a few hundred kW to 5 MW. The smallest DG units commercially available today can produce 30 kW.

Image 1 - Distributed Generation Process

1.3 Dissertation Overview

The first chapter gave a brief context of what is Distributed Generation and the company where this project was built, EDF Energy Networks. You can see also in this chapter the problem description that contains a description of what is the problem with the Distributed Generation at the moment in EDF Energy Networks, introducing the essential concepts which are explained later in the report.

Chapter 2 (Approach for Specifying the DGIS) gives a theoretical approach about Software Requirements Specification, Human-Computer Interaction, the platforms that could be used to do this Information System and how the project was initially planned.

Chapter 3 (Proposed IS for DG) shows what was the requirement specification done and the respective prototype.

Chapter 4 (Evaluation of proposed IS) gives the evaluation of the requirement specification and prototype done, based on the theoretical approach done in Chapter 2

Chapter 5 (Conclusion and Future Work) presents some conclusions and gives some hints to finalize this project.

1.4 Problem Description

EDF Energy Networks is divided in 3 licensed areas at the moment SPN, EPN and LPN. Each of these areas has their own way to manage the Generation of Distributed Energy and there is no main/single database that holds all the required information. They usually do it in Excel sheets, Access databases and Word documents.

Image 2 - Companies managing electrical distribution on UK

When someone needs a report regarding Distributed Generation it takes a lot of time to get it and normally, when it comes out, it's useful and no one needs it anymore. This happens because the report depends on the responsible of the corresponding area (sometimes on holidays) and depends also on the complexity of the report.

Another problem with the current DG system is that it has a lot of wrong data because it's not updated. Sometimes the data that is on the system doesn't correspond to the true, for example, customer addresses, sites that don't exist anymore, etc.

The other issue is that the stored data has a lot of inconsistency, for example, if a site has 2 generators with an output of 5kW each, the site has a total output of 10kW and sometimes this doesn't happen.

The main achievement of this information system will be the considerably time reduction of report production, the common information in all of the 3 licensed areas (SPN, EPN and LPN), the accessibility from one source and the removal of inconsistency.

Chapter 2

Approach for Specifying the DGIS

In this chapter it's told what were the methodologies used to do this information system, in particular, the software requirement specification, the best way to design the interface (human computer interaction), the choice of used platforms and what were the project priorities and plans.

2.1 Software Requirement Specification

The objective of the specification of the software requirements is describing, with as much detail as possible, the requirements that this IS needs to have and also what kind of actors will be using this system. This is a fundamental tool because of the following reasons:

1. Allows EDF Energy Networks to confirm the with detail and accuracy the proposed requirements and suggest some changes
2. Allows the project manager to better make a budget of the system and plan all the process of development
3. Helps the supervisors to analyze and check if each use case corresponds to the functions in the system.
4. Helps the developers to understand with better efficiency and accuracy what exactly the system needs to do.
5. Helps the development of the necessary tests to the IS to validate each of the requirements.
6. Allows the maintenance team to better understand the system.

Before I started the Software Requirement Specification I made some research to manage the best way to achieve the Software Requirement Specification report structure. I started to ask Prof. Ademar Aguiar from FEUP for a model of a Software Requirement Specification Report and together with the norm 830-1998 of IEEE (*IEEE Recommended Practice for Software Requirements Specification)* I made the Software Requirement Specification Report that is on Appendix A with the following structure [IEEE09, AA02]:

- Introduction
    - Objective
    - Scope
    - Structure of the document

- General Description
    - Product Context
    - Functionalities of the product
    - User Characteristics
    - Assumptions and dependencies
- Specific requirements
    - Requirements for external interfaces
    - Functional Requirements
    - Use cases
    - Non-functional Requirements
    - Information Requirements
    - Glossary
    - References

Each Use Case is described with the following structure:

Table 1 - Use Case Structure

Identifier	Short name to allow the brief uniquely identification of the use cases
Name	Long name to give a better understanding of the use case
Brief description	One or two short sentences making clear the purpose / utility of the use case
Actors	Give the various stakeholders and their role, making clear who initiates the interaction. (These actors are previously described on the previous chapter on the report)
Priority of implementation	This establishes a priority to the implementation of the use case (essential, desirable, optional) for development purposes.
Sequence of operation / flow of events	This part describes the normal, alternative and exceptional sequence of operations. Indicates the actions performed by the actors and the system, making clear the input data (inserted by the actors) and output (provided by the system). It needs to be clear also how the use case begins. For the better understanding the most complex use cases have diagrams of activity to facilitate the communication and remove ambiguities.
Interface with user	In this section it is shown pictures of the prototype of the interface with the user.
Pre-conditions and restrictions	This section shows the restrictions and pre-conditions in the input data and initial state of the system
Post-conditions	A post-condition is a condition that relates the data output and final state of the system with data input and initial state of the system. It reflects the effect/outcome of the case of use.
Main scenario	In this section is specified what is the most probable scenario that happens on the sequence of operations.
Alternative scenario	In this section is specified what is the less probable scenario that happens on the

	sequence of operations. Some use cases do not have alternative scenarios
Frequency of use	The frequency of use specifies from 1 to 10 if the actors will use this function very often, 10 are the most used.

2.2 Human Computer Interaction

"*Human-Computer Interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.*" – ACM SIGCHI (2002)

Before starting with the drawing of the sketches some questions like the following had to be answered:

- What's the problem?
- What are the user needs?
- What are the objectives of usability?
- What do we want to create?
- What are the assumptions?
- It will achieve what is needed?

The risk associated to a useful system is a system that doesn't correspond to the expectations and more time and money has to be spent. So to answer to all of these previous questions I started to do some research as you can see next about Human-Computer Interaction after doing the Software Requirement Specification report. [WikiI09, PREECE02]

2.2.1 Overview

The project of creating an interface is an evolutive process that has the objective of achieving a usable project (easy to learn, effective and that gives an enjoyable experience). During the process of creation the users have to be involved during all the process. The type of user, type of activity and the context of interaction have to be also taken in account.

It's easy to learn doing and repeating the process so there was a preoccupation of maintaining the process on every stage of the interface as you can see in the next chapter. [WikiU09, WikiH09]

2.2.2 Nielsen Heuristics

During the development of the prototype the 10 heuristics of Nielsen were taken in to account as you can see it applied on the next chapter. The 10 heuristics of Nielsen are the following [NIELSEN09]:

1. <u>Visibility of system status</u> - The system should always keep the users informed about what is going on, through appropriate feedback within reasonable time.
2. <u>Match between system and the real world</u> - The system should speak the user's language, with word, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
3. <u>User control and freedom</u> - Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

4. Consistency and standards - Users should not have to wonder whether different words, situations or actions mean the same thing. Follow platform conventions.
5. Error prevention - Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
6. Recognition rather than recall - Make objects, actions and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
7. Flexibility and efficiency of use - Accelerators - unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
8. Aesthetic and minimalist design - Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialog completes with the relevant units of information and diminishes their relative visibility.
9. Help users recognize, diagnose and recover from errors - Error messages should be expressed in plain language (no codes), precisely indicate the problem and constructively suggest a solution.
10. Help and documentation - Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out and not be too large.

2.2.3 Three Principles of Shneiderman

Ben Shneiderman, human computer scientist and professor of Computer Science at Human-Computer Interaction Laboratory at the University of Maryland conducted fundamental research in the field of human–computer interaction, developing new ideas, methods and tools such as the direct manipulation interface and created the following three rules that will be applied to the interface of this project[Shnei04, WikiS09]:

A. Recognize the diversity: In order to recognize diversity, it has to be taken into account the type of user frequenting the system, ranging from novice user, knowledgeable but intermittent user and expert frequent user. Each type of user expects the screen layout to accommodate their desires, novices needing extensive help, experts wanting to get where they want to go as quickly as possible. Accommodating both styles on the same page can be quite challenging. You can address the differences in users by including both menu or icon choices as well as commands or providing an option for both full descriptive menus and single letter commands.

B. Use the Eight Golden Rules of Interface Design
    1. Strive for consistency: Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus and help screens: and consistent commands should be employed throughout.
    2. Enable frequent users to use shortcuts: As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands and macro facilities are very helpful to an expert user.
    3. Offer informative feedback: For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be substantial.
    4. Design dialogs to yield closure: Sequences of actions should be organized into groups with a beginning, middle and end. The informative feedback at the

completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds and an indication that the ay is clear to prepare for the next group of actions.
5. Offer error prevention and simple error handling: As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.
6. Permit easy reversal of actions: This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry or a complete group of actions.
7. Support internal locus of control: Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.
8. Reduce short-term memory load: The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced and sufficient training time be allotted for codes, mnemonics and sequence of actions.

C. Prevent Errors

2.2.4 George Miller Theory

"The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information" – George A. Miller (Department of Psychology, Princeton University)

This is one of the most highly cited papers in psychology and is also applied in Human-Computer Interaction. Briefly, the paper says that interesting coincidences existed between the channel capacity of a number of human cognitive and perceptual tasks. In each case, the effective channel capacity is equivalent to between 5 and 9 equally-weighted error-less choices: on average, about 2.5 bits of information. In other words, it's considered that working memory has a limited capacity and with some measurements of human short term memory capacity Miller also found a 7±2 limit capacity.

So, in interface design the developers have to be sure that the buttons, menu options, etc are between 5 and 9 items when possible, for a better user memorisation [MILLER56].

2.2.5 Evaluate and test the interface

To evaluate and test the final interface I used informal meeting with the users, tests of usability and interviews with the final users to evaluate their level of satisfaction based on the following points [PREECE02, NORMAN98]:

- Effectiveness
- Efficiency
- Security
- Utility
- Learning
- Utility
- Memory of how the system works

- Satisfaction
- Fun
- Aesthetic beauty
- Motivation
- Help feeling

2.3 Platform

When this internship started EDF Energy Networks didn't knew which technology should have to be followed. So I started to separate the Web applications from Desktop Applications and after that decision we could decide which technology should we use, based on the existing licensed technologies installed inside the company.

Web Application:

Normally is more used in Internet because of the widespread of the web br owsers as client but it can be used in Intranet also.

Pros:

- Easily accessible from any PC and any location with Intranet or Internet access.
- Requires a single installation on the server.
- Very easy to update in the server instead of each client.

Cons:

- The web applications tend to be slower in the network because has to transfer a lot more information.

Desktop Application:

The front end runs in the local machine and it just have to exchange the data from the server and not the interface itself.

Pros:

- Faster than web application because has less information transferred.

Cons:

- Unless there is a server with the application running inside the network (remote control) the desktop application have to be individually installed.
- If there is an update and the application is installed in every computer then it has to be updated in every PC.

After the decision of developing the Information System into a Web Application the chosen technology was J2EE to the front-end together with an Oracle database to store the data.

2.4 Project Priorities and Plans

Usually, the software projects are divided in the following 6 stages [WIKISD09, WIKIR09]:

1. Requirement analysis and planning: This stage determines identifies the needs and conditions to meet for a new product and define the requirements, taking in account the conflicted requirements of the various stakeholders.
2. Design: The objective of this stage is to draft the first piece of software to allow the developer to have more knowledge about the subject area, Distributed Generation in this case, and to remove any possible confusion of the 2 parts

(developer and EDF Energy Networks) of what should be the final Information System.

3. Specification: In this stage the main achievement is to specify and describe precisely what the software should do. This allows the developer to make sure that the software goes inline that with what was proposed to do and all the other reasons that are described on the beginning of chapter 2.1.
4. Architecture: This is an abstract representation of the system to make sure that the system meets the requirements of the product and that the future requirements can be addressed. This is normally represented with a final prototype.
5. Implementation, testing and documenting: This is the part where the engineers code the program specified in the previous stages, do the tests also with the help of the specification and documentation if necessary.
6. Deployment and maintenance: After the testing of the program and approved for release, this is the stage responsible for maintaining the program working and for the correction of some probable bug.

For the time being of this internship of 4 months only the first 4 steps were planed. The Gant chart of the planned activities is on Appendix G.

2.4.1 Software Development Methodology

For this project the model used to develop this project was **Rapid Application Development.** [WIKIRAD09, WEBN09].

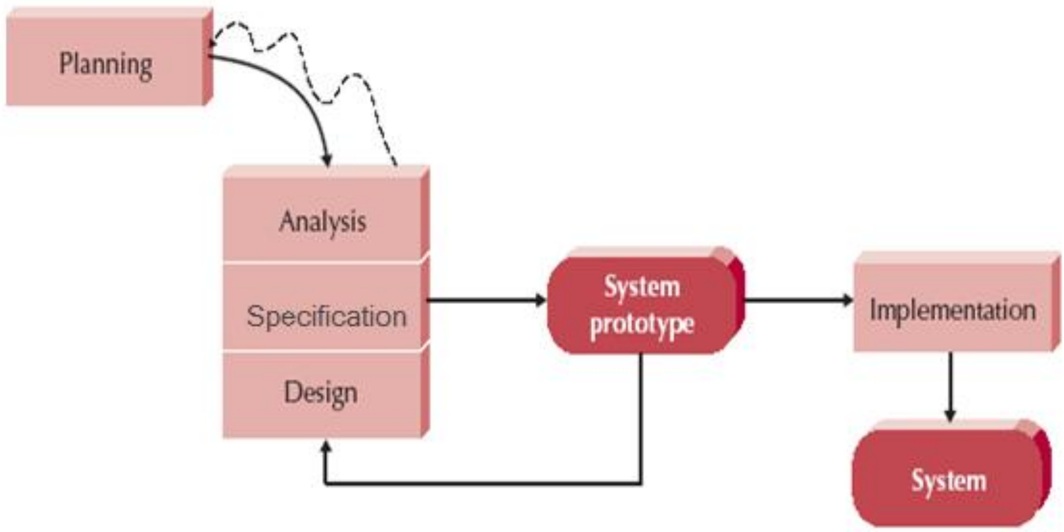

Image 3 - Rapid Application Development

This kind of development requires minimal planning in favour of rapid prototyping that is actually what is wanted in this project because as developer I don't have much information about the constraints and scope of the project, the only constraint that I've had was the 4 months of internship. RAD also allowed regularly feedback from the users and managed to do a better specification and a better idea of what would be the final interface achievement.

On the first stage, the Planning, I couldn't do a very good approach of the time that the tasks would take because I didn't knew the total scope of the project and I don't had a lot of experience on the planning of this kind of projects. The only constraint that I had was that this project needs to have 4 months of duration that corresponds to the time of the internship. So I decided to divide the project in 3 “loops” in RAD. The first loop would be the first approach that would allow me to do the first sketches on paper, some requirement analysis but still with limited understanding and only with a base overview about the scope structure. The second stage would be to achieve a MS Access prototype, now with limits on scope, which would allow me to get better feedback from a greater audience of users. And finally, the web based iteration, with a total understanding of the project scope, which allowed me to have the feedback from almost all users.

Analysis, Specification and Design, is the stage where the system requirements have to be raised, reported and from those requirements a prototype is built to get feedback from the users to go through a new loop. To see more about the planning of the planning of this

Chapter 3

Proposed IS for DG

In this chapter it is explained how the approach for specifying the Information System, in the previous chapter, is applied to this Distributed Generation Project.

3.1 Requirements – an Overview

Software Requirement Specification is a complete description of the actions of the system to be developed. It includes a set of use cases or functional requirements that describes all the interaction between the users and the system and the non-functional requirements that impose some constrains on the design and implementation. The Software Requirement Specification Report that can be seen on Appendix A of this report.

To achieve the final objective of this stage (SRS Report on Appendix A) I started to do an excel sheet to specify what were the fields that this IS has to store based on the forms that the customers send to EDF Energy Networks to install is owns generation and from existing excel sheets, word documents and other databases on the different areas of the Network (SPN, LPN, EPN licensed areas). The excel sheet includes the dependencies between data, examples, type of data, which data is compulsory and if it has any consistency check (for example data has to be in dd/mm/yyyy format). After the conclusion of the excel sheet I made an enquiry to the company (Appendix B) to better understand the terms related to Distributed Generation to construct the Entity-Relationship Model and the interface prototypes.

1 BASIC INFORMATION OK

1,1 Owner Details

Type	Field ID	Field Name	Example	Data Format	Consistency check	Alert	Input (Y/N)	Compulsory (Y/N)	Relation
		Owner's Address							
C	1	Name	Renewable Zukunft Ltd	text	None	None	Y	1	Owner
C	2	Address	Mill Yard	text	None	None	Y	2	Owner
C	3	Town	Childerley Estate	text	None	None	Y	2	Owner
C	4								
C	5	Postcode	CB23 8BA	text	Max = 4 + 3	None	Y	2	Owner
C	6	Contact Name	Tim Evans	text	None	None	Y	2	Owner
C	7	Phone Number	01284726793	number	11 digits	None	Y	2	Owner
C	8	Mobile Number	07875111117	number	11 digits	None	Y		Owner
C	9	Fax Number	08701962255	number	11 digits	None	Y		Owner
C	10	Email	cristiano.marantes@edfenergy.com	text	"@" and "."	None	Y		Owner
		Billing Address							
G-59	11	Name	HoS plc	text	None	If same use info above (?)	Y	2	Owner
G-59	12	Address	15 Regency Square	text	None	If same use info above (?)	Y	2	Owner
G-59	13	Town	Cambridge	text	None	If same use info above (?)	N	2	Owner
G-59	14								
G-59	15	Postcode	CB1 3WL	text	Max = 4 + 3	If same use info above (?)	Y	2	Owner
G-59	16	Contact Name	Cristiano Marantes	text	None	If same use info above (?)	Y	2	Owner
G-59	17	Phone Number	01284726793	number	11 digits	If same use info above (?)	Y	2	Owner
G-59	18	Mobile Number	07875111117	number	11 digits	If same use info above (?)	Y		Owner
G-59	19	Fax Number	08701962255	number	11 digits	If same use info above (?)	Y		Owner
G-59	20	Email	cristiano.marantes@edfenergy.com	text	"@" and "."	If same use info above (?)	Y		Owner
		Refund Address							
G-59	21	Name	HoS plc	text	None	If same use info above (?)	Y	2	Owner
G-59	22	Address	15 Regency Square	text	None	If same use info above (?)	Y	2	Owner
G-59	23	Town	Cambridge	text	None	If same use info above (?)	N	2	Owner
G-59	24								
G-59	25	Postcode	CB1 3WL	text	Max = 4 + 3	If same use info above (?)	Y	2	Owner
G-59	26	Contact Name	Cristiano Marantes	text	None	If same use info above (?)	Y	2	Owner
G-59	27	Phone Number	01284726793	number	11 digits	If same use info above (?)	Y	2	Owner
G-59	28	Mobile Number	07875111117	number	11 digits	If same use info above (?)	Y		Owner
G-59	29	Fax Number	08701962255	number	11 digits	If same use info above (?)	Y		Owner
G-59	30	Email	cristiano.marantes@edfenergy.com	text	"@" and "."	If same use info above (?)	Y		Owner
C	31	Comments /Notes	Owner is Equipment owner and not the land owner	text	None		Y		Owner

Image 4 - Example of excel sheet page

In the Image 4 you can see an example of the Owner Details on excel sheet. The first field, type, indicates if the field is Common, is only from G-59 or G-83 Form. The second, Filed ID, indentifies uniquely the field, the third Field Name, says what is the name of the field, the fourth field has an example of data, the fifth has the data format (text, number, data, etc), the sixth has a consistency check to avoid mistypes, the seventh is an alert for the developer, the eighth is to allow the developer to know if the field is an input or if is an output from the system, the ninth indicates if the field is compulsory on the new application, on a later stage or if it's not compulsory and finally the tenth indicates the relation of the field to the entities to better understand the system and construct the E-R model.

In the beginning EDF Energy Networks had an abstract idea of what they want as an end result. Incomplete, ambiguous and contradictory requirements were recognized in this stage.

After the conclusion of the excel sheet and for the better understanding of the users I started to do some prototypes on paper that was shown and approved to a small group of managers. After that an Access (Image 5) prototype was done to get more opinions of a larger audience and after some discussion we realize that the process that Distributed Generation takes inside the company has to be reflected in the Interface, so a Flowchart was done (Appendix C) to better clarify where and by whom the Information System would be updated. After some decisions and a better clarification on this matter the prototype was upgraded to a Web based technology where usability and page flow questions were discussed, as you can see in the next chapter, with a large audience in all licensed areas.

3.2 User Interface

To achieve the final prototype of the user interface there was an evolution process that begun with drafts on paper that are not included on this reports because of logistic issues, after that an MS Access interface was built (Appendix E and Image 5) and, finally a final web based interface was built that can be seen on the (Appendix F or on Software Requirement Specification Report on Appendix A). All of this 3 "Interfaces" were getting better and better with the opinions of the final users together with the discussion with them based on the knowledge that was given to me in Human-Computer course.

To achieve the final interface (Image 6), ideas were taken of SiFEUP and from the current intranet of the company. This method of extracting ideas and processes from the company intranet allowed the user to, intuitively, know how to manage with the system.

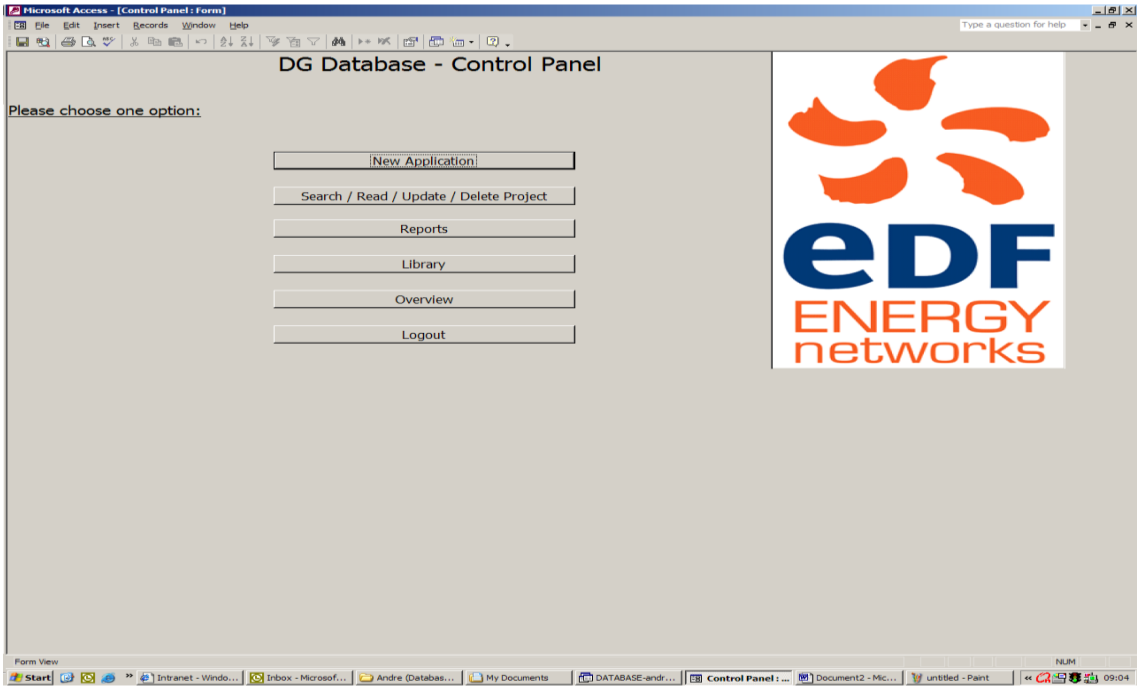

Image 5 - Access Prototype Screenshot

Chapter 4

Evaluation of proposed IS

4.1 Requirements

Most of the bugs in software are due to incomplete and inaccurate functional requirements. So it's better to catch the requirement ambiguities and fix them in early development life cycle to avoid the extremely high cost of fixing the bug after the completion of the development or release of the product.

Before the final release of the final SRS Report it was checked that all the requirements are clear and consistent and to prove that the raised requirements were correctly specified some tests with the users were done to the prototype with the insertion of some G59, G83 and Multiple G83 forms.

4.2 User Interface

To prove that the achieved interface goes to the encounter to what was said on chapter 2.2 it will be shown some interfaces and it will be explained on the context of the human-computer interaction.

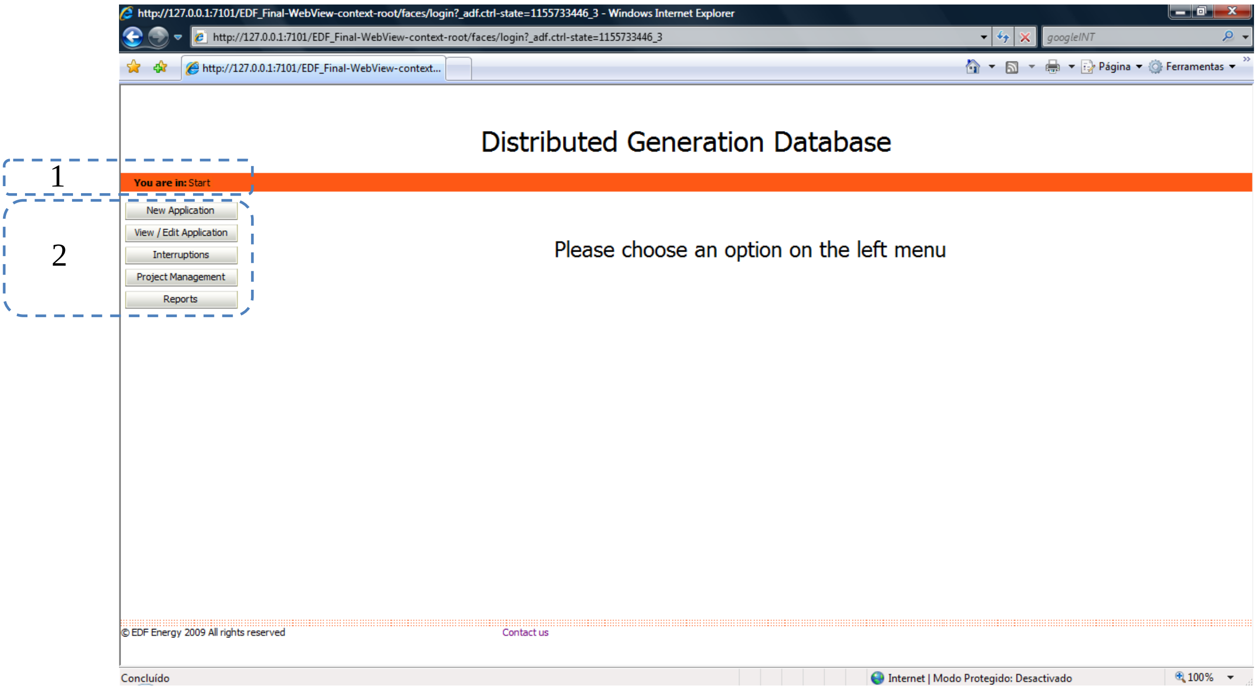

Image 6- Initial page of the Proposed IS

On the Image 6- Initial page of the Proposed IS we can see proved the following points:

1. The section marked as 1 on the interface keeps the user informed where it is and what is going on so the 1st Nielsen Heuristic is present over all the interfaces on this IS.
2. The section marked as 2 on the interface is the main menu also present over all interfaces. It passes on the following tests in Human-Computer Interaction:
   - George Miller Theory with 5 buttons (sometimes 6 it depends if it's the administrator logged or a normal user). This allows the better memorisation of the user.
   - 3rd heuristic of Nielsen, User control and freedom, allows the user to go back to any menu of his choice anywhere in the system.
   - 6th heuristic of Nielsen, Recognition rather than recall, this menu is always visible and allows the user to do not have to remember the menu.

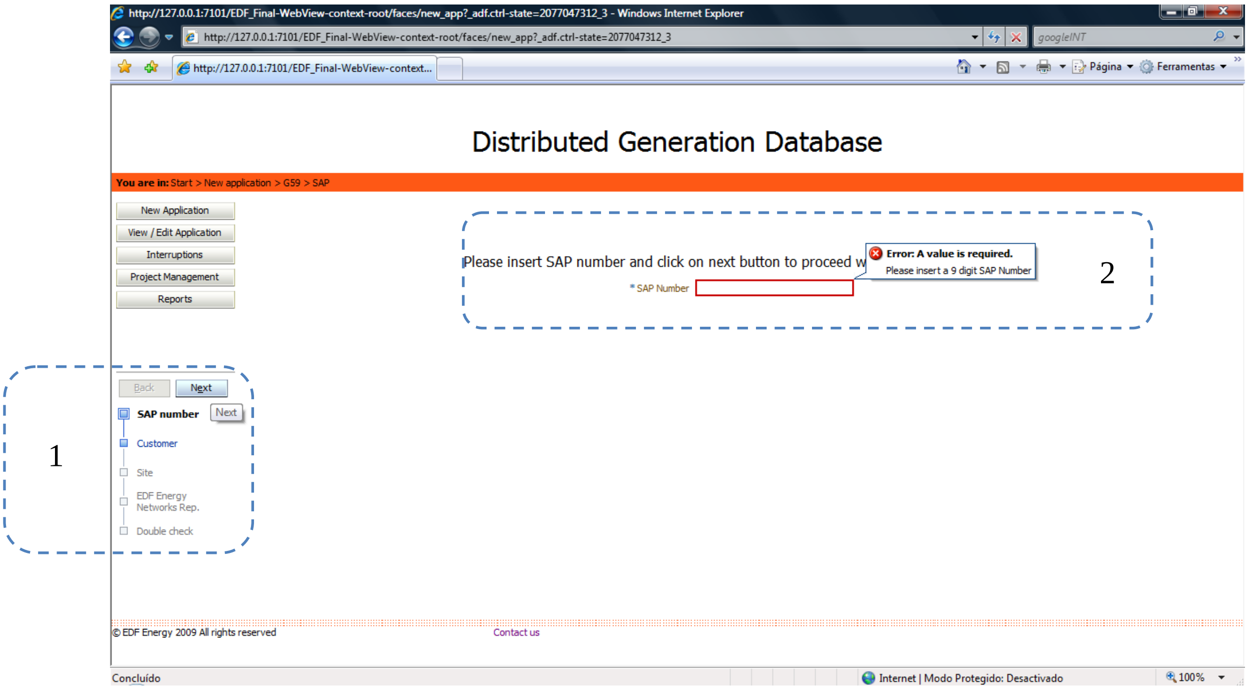

Image 7 - New G59 Application Interface

On the Image 7 - New G59 Application Interface of the proposed IS the previous proved points remain in the interface and we also can prove the following points:

1. On the section marked as 1 the George Miller theory continues present with 5 options on the menu. The gray options indicates that the user cannot go directly those options, for example, he can't go to Site without filling SAP number. The bold (SAP number) section indicates where the user is together with the status bar. The Owner is blue because indicates the next step and that is clickable. As we will see in the next picture, a lilac color indicates that the stage was accepted by the system but the user can go back to it if he wants to.
2. On the section marked as 2 the system is giving an error because the user tried to proceed to the next step without filling the SAP Number that is compulsory. The 8th and 9th Nielsen heuristic is proved here because the interface has Aesthetic and minimalist design and the error message recognized the user recognizing, diagnose and recover from the error.

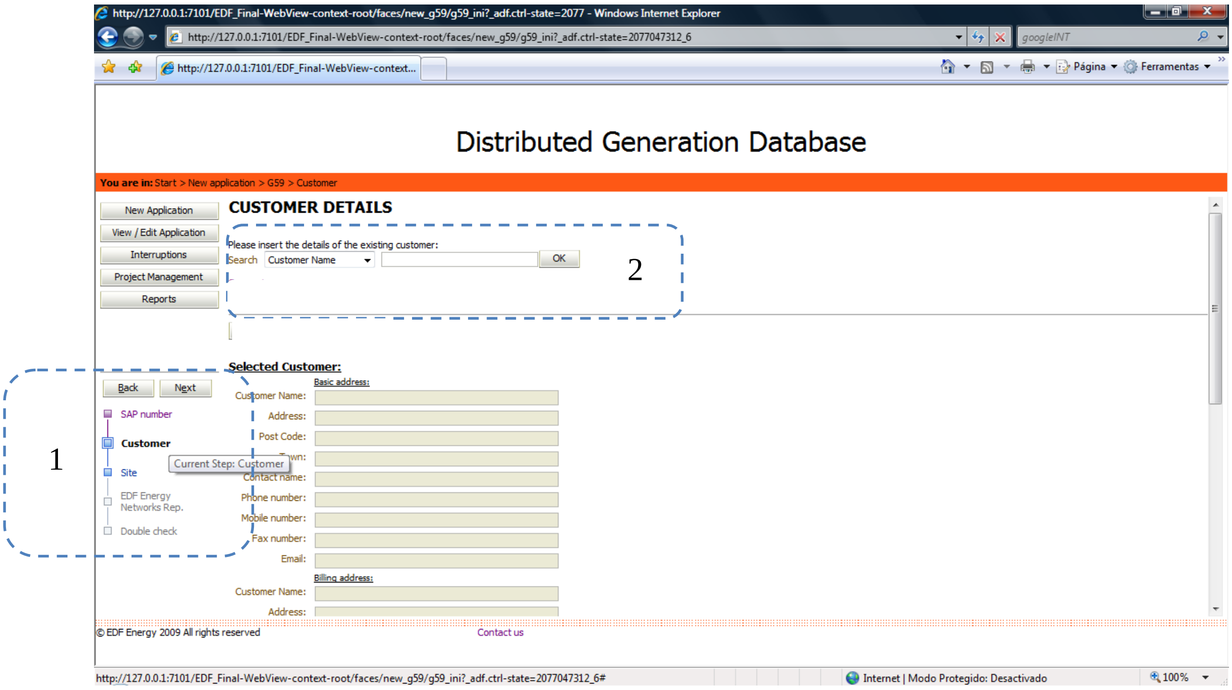

Image 8 - New G59 Application Customer Section

On the Image 8 - New G59 Application Customer Section of the proposed IS the previous proved points remain in the interface and we also can prove the following points:

1. On the section marked with 1 the user can easily see what steps he completed or visited, the current step in bold, and the next step he can go. He can also see all the remaining steps to complete the application, even without the ability to click on them. Also on section 1 you can see that the user can mouse-over any button to get additional help. The 1st, 3rd, 7th, 9th, 10th are proven in this section together with the three principles of Shneiderman and George Miller.
2. On the section marked with 2 the user needs to search for a customer before creating a new one to avoid duplication of incorrect data. So the option of creating a new customer will be shown after the search. The only thing that the user can fill in this page is the search box, everything else is disabled to avoid confusion and get more simplicity. This process of search before creating is maintained through the system to a better consistency and efficiency of use, you can see another example in Image 9.

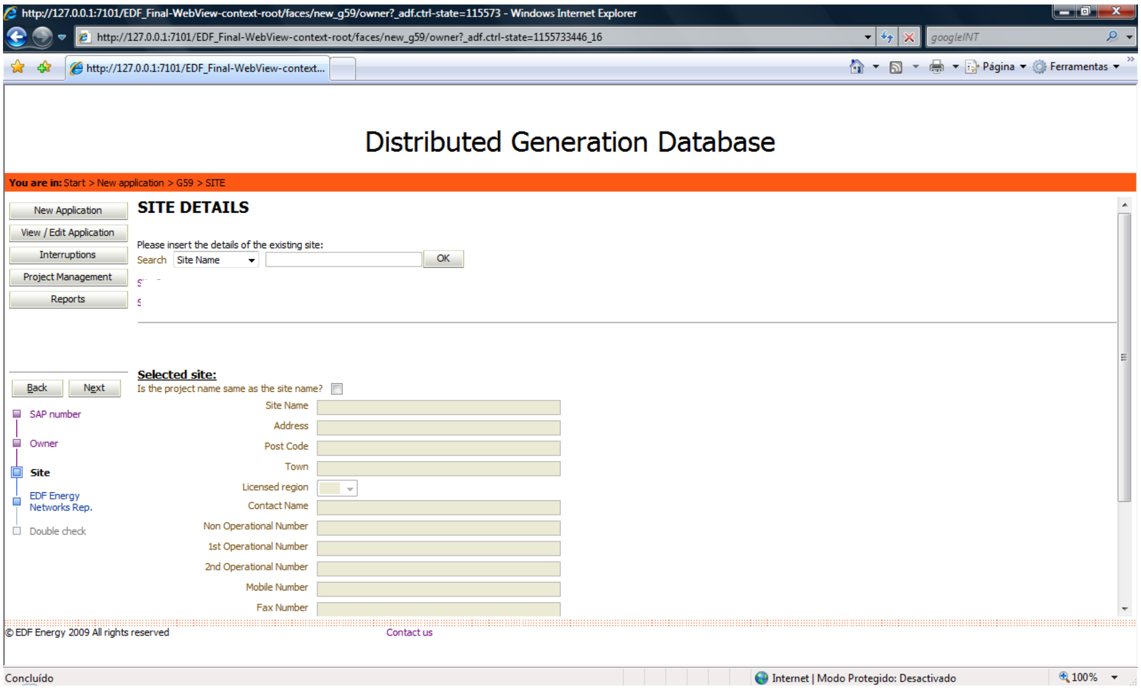

Image 9 - New G59 Application Site Section

The shown interfaces were all related to a new application. Now it will be showed the View / Edit Application.

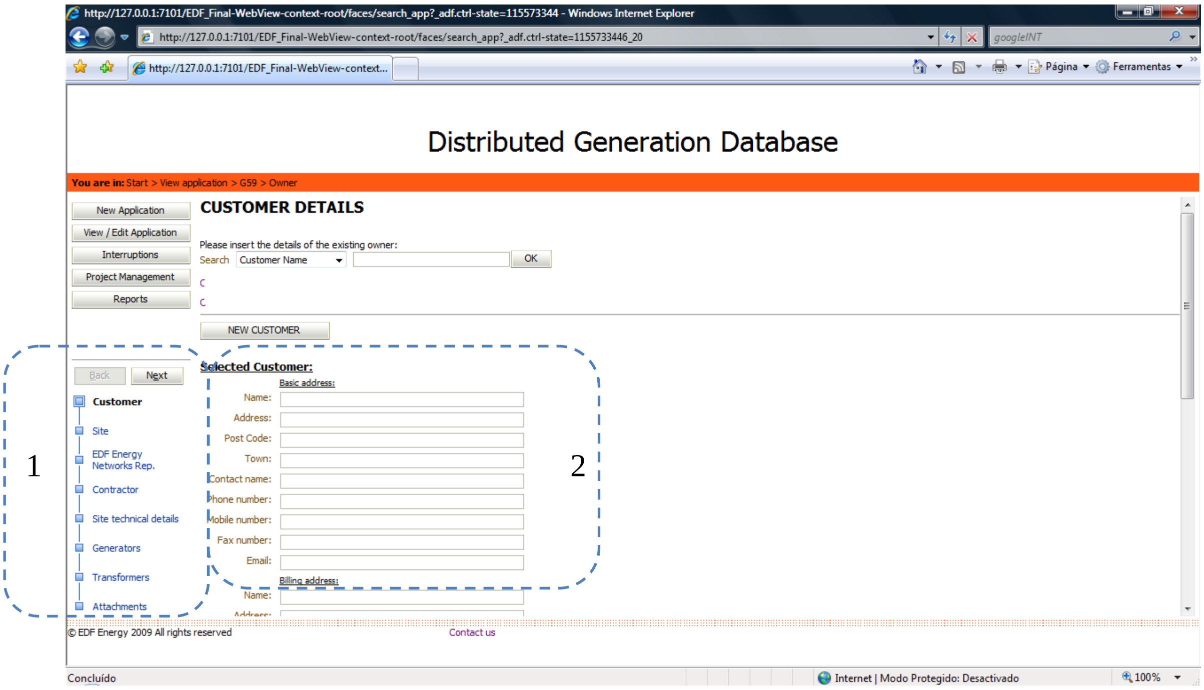

Image 10 - View/Edit G59 Application Customer Edition

The process on the View / Edit application is pretty much the same of the New Application the main differences is that now the user can jump directly to the step he wants (Image 10 - Mark 1) and he can edit the customer details directly (Image 10 - Mark 2).

In the next steps it will be shown the screenshots of the more important parts of the system and a general explanation of why the interface meets the principles of Human Computer Interaction (2.2 - Human Computer Interaction). If you need more detail about the interface you can see the Appendix F.

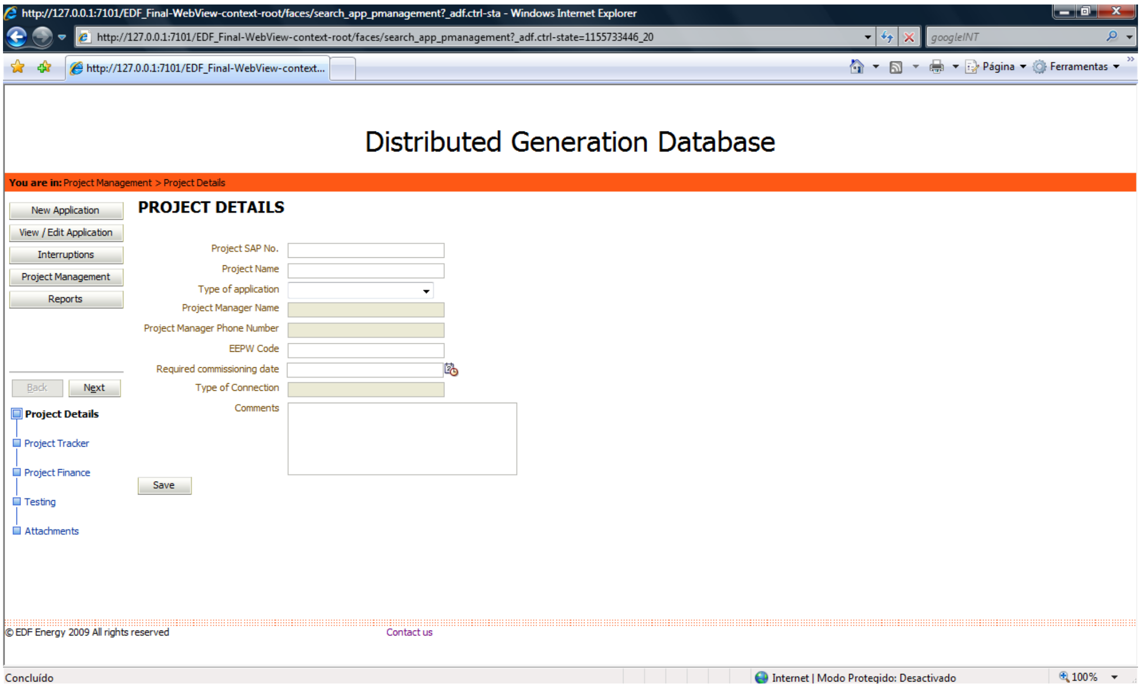

Image 11 - Project Management Interface

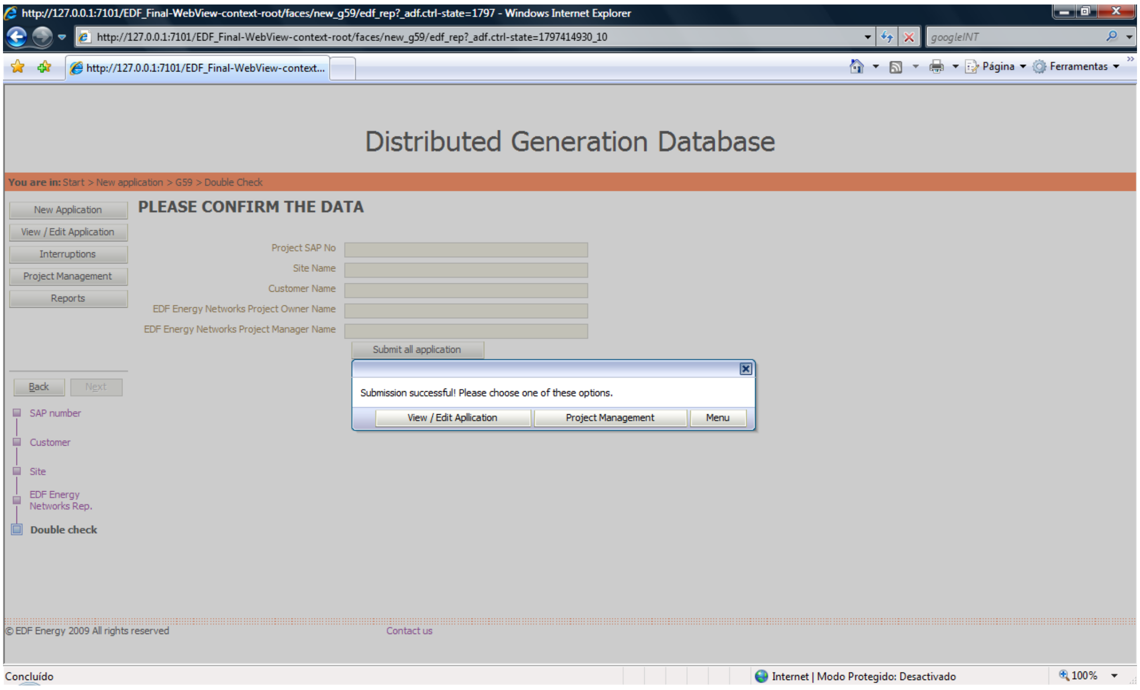

Image 12 - G59 Application Successfully Submission

4.2.1 Nielson Heuristics

In this section it will be explained how I applied the Nielsen Heuristics to the interface prototype (Appendix F).

1. Visibility of system status – There is a status bar present in all stages of the interface that orientates the user telling him where is he (Mark 1 of Image 6).
2. Match between system and the real world – The target users are electrical engineers and technicians so all the terms used in this IS are familiar to the users.
3. User control and freedom – In case of a mistype or miss click the user can always go back to the previous stage (Mark 1 of Image 8 and Image 10) or search and select a different entity (Mark 2 of Image 8)
4. Consistency and standards – The IS tries to follow the same patterns and procedures with the way that the user navigates in the system (Mark 1 of Image 8 and Image 10) and the way the user searches and selects an entity (Mark 2 of Image 8).
5. Error prevention – There was a concern about leaving the interface simple and clean with brief explanations of what the user has to do to prevent errors as you can see it.
6. Recognition rather than recall – Navigation buttons of the overall system (Mark 2 of Image 6) or in the application (Mark 1 of Image 7) are always visible. When the user needs help he just have to mouse over the buttons he wants for the help to be showed (Mark 1 of Image 8).
7. Flexibility and efficiency of use – In a View / Edit application there are shortcuts directly for the section that the user instead of clicking on "Next" button to get there (Mark 1 of Image 10). This doesn't happen with a new application because all the stages are compulsory, so the user can go back but it cannot go further more than 1 step (Mark 1 of Image 8).
8. Aesthetic and minimalist design – As you can see on the shown interfaces shown before or on (Appendix F) this interface presents a very aesthetic, simple, clean and minimalist design.
9. Help users recognize, diagnose and recover from errors – As you can see (Mark 2 of Image 7) the errors are indicated exactly in the place where they happen, rather than a popup saying that there is an error on the interface, and the error indicates clearly what the user must do.
10. Help and documentation – An intuitive and easy to learn system doesn't need help menus but I decided to include the help with the mouse over of the functions rather than create a user manual that the target users won't have time to read.

4.2.2 Three Principles of Shneiderman

A. Recognize the diversity: One of the first priorities of this project was the raise of all the possible type of users in order to design the interface and build the use cases.

B. Use de Eight Golden Rules of Interface Design
   1. Strive for consistency: As I said before in the 4th Nielson Heuristic the consistency is present in all actions, prompts and help screens.
   2. Enable frequent users to use shortcuts: This is also pretty much the same rule as the 7th Nielsen Heuristic that is present through the interface as I explained before.
   3. Offer informative feedback: Anytime that the user makes an error the system gives the respective feedback (Mark 2 of Image 7) and when it finishes some action he also receives feedback as you can see in Image 12 when it finishes the submission of an application.

4. Design dialogs to yield closure: As you can see the creation of an application is divided in several steps (Mark 1 of Image 8) with a beginning, middle and end (Double check) and in the end the sequence of operations has a informative feedback (Image 12).
5. Offer error prevention and simple error handling: This golden rule is the same as the 5th Nielson Heuristic explained before.
6. Permit easy reversal of actions: There was a concern about leaving the interface simple and clean with brief explanations of what the user has to do to prevent errors and in case of a mistype or miss click the user can always go back to the previous stage (Mark 1 of Image 8 and Image 10) or search and select a different entity (Mark 2 of Image 8).
7. Support internal locus of control: I based myself in the accomplishment of this rule based in the feedback that I had from the users
8. Reduce short-term memory load: To achieve this golden rule I followed George Miller theory creating not more than 9 buttons or stages and not less than 5.

C. Prevent Errors – As said before the produced aesthetic, simple, clean and minimalist interface was designed to avoid and prevent errors.

4.2.3 George Miller Theory

As you can see along the final interface (Appendix F) there was a preoccupation of not showing more than 9 and less than 5 buttons, stages, menu options, etc in the same page whenever was possible.

Chapter 5

Conclusion and Future Work

"Success is a journey, not a destination."
- Ben Sweetland

This chapter serves as a conclusion to the overall subject covered throughout this report and delineates the expected further developments to this project.

5.1 Conclusion

Throughout the duration of the DG Information System project, it as possible to go through all the phases of the specification of a software project, surpassing technical and social challenges that were essential to the good conclusion of the project. This dissertation is the most important and, side by side with the system that was produced, the most tangible result of the project.

The proposed software specification of the system and its respective interface were successfully achieved during the internship. All the early support and knowledge provided by everyone at EDF Energy Networks helped keeping this traineeship in the right track. The developed interface prototype was also very handy as a first sketch of what the final implementation should look like.

Despite being a one-person team, the team devoted to the project required management: by the supervisors at EDF Energy and FEUP devoted to this project, but also, first and foremost throughout the project, by the author of the project. Managing time, project scope, external expectations and participatory management in the planning, execution and promotion of the prototypes was essential.

Both main requirements (Prototype and Software Requirement Specification Report) for this project were accomplished with the agreement of all the stakeholders of this project making a step forward to this project. One of the difficult issues in the development of this project was to get the feedback from the key users (project managers) that have a very busy agenda and sometimes were also on holidays which triggered sometimes a delay on the approval and in the development of the stages.

Working in a company with the dimension of EDF Energy was a very enriching experience. This was due to the fact that it has a large cultural diversity and different ways of thinking which opened my mind during the development of this project. Also the mix of cultures allowed me to improve my communication skills with people from different technical backgrounds. The experience also enabled me to dismiss any pre-conceived racial and ethnic stereotypes and has given me the opportunity to explore into the historical and cultural insight knowledge of the different corners of the world.

5.2 Future Work

Software development easily attracts a desire for constant and never ending improvement. It is of upmost difficulty to establish the real end of any software project, since requirements always change along the way, there always something to fix and features that always be improved to work smoother or simply more elegantly. In the end of this project the specification and prototype had the approval of all stakeholders so it can be said that reached a good quality of excellence.

Now that the complete software specification and prototype is done I am very gratifying to know that the developed solution is going to be further enriched with the implementation of the Information System in the Intranet of EDF Energy Networks that was designed in this project.

References

[AGUIAR09] Ademar Aguiar. Software Requirement Specification Template Report. Faculdade de Engenharia da Universidade do Porto, June 2009.

[EDFG09] EDF Group. EDF Group Website, June 2009. http://www.edf.com/

[EDFN09] EDF Energy Networks. EDF Energy Networks Website, June 2009. http://www.edfenergy.com

[IEEE09] IEEE. IEEE recommended practice for software requirements specifications 830-1998 Norm, June 2009. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=720574&isnumber=15571

[MILLER56] George A. Miller, The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information, 1956, Psychological Review.

[NG09] National Grid. Distribution Network Operator (DNO) Companies, June 2009. http://www.nationalgrid.com/uk/Electricity/AboutElectricity/DistributionCompanies/

[NIELSEN09] Jakob Nielsen. Ten Usability Heuristics for user Interface Design, June 2009. http://www.useit.com/papers/heuristic/heuristic_list.html

[NORMAN98] Donald A. Norman: The Invisible Computer, 3rd printing, 1999, Cambridge, Massachusetts: The MIT Press 1998, ISBN: 0262140659. www.jnd.org

[PREECE02] Jenny Preece, Yvonne Rogers, Helen Sharp: Interaction Design: Beyond Human-Computer Interaction, 2nd edition, January 2002. ISBN: 0470018666. www.id-book.com

[SHNEI04] Shneiderman Shneiderman, B. (2004). (4th ed.), Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison- Wesley Publishers: Reading, MA

[WEBN09] Webnetters.org. Rapid Application Development – introduction, June 2009 http://forum.webnetters.org/international-forum/rapid-application-3318.html

[WIKIH09] Wikipedia. Human-Computer Interaction, June 2009. http://en.wikipedia.org/wiki/Human-computer_interaction

[WIKII09] Wikipedia, Interaction Design, June 2009. http://en.wikipedia.org/wiki/Interaction_design

[WIKIR09] Wikipedia. Requirement analysis, June 2009 http://en.wikipedia.org/wiki/Requirements_analysis

[WIKIRAD09] Wikipedia. Rapid Application Development, June 2009 http://en.wikipedia.org/wiki/Rapid_application_development

[WIKIS09] Wikipedia. Ben Shneiderman, June 2009 http://en.wikipedia.org/wiki/Ben_Shneiderman

[WIKISD09] Wikipedia. Software Development Process, June2009 http://en.wikipedia.org/wiki/Software_development_process

[WIKIU09] Wikipedia. Usability, June 2009. http://en.wikipedia.org/wiki/Usability

Appendix A

Software Requirement Specification Report

Appendix B

EDF Energy Networks Inquiry

Appendix C

Flowcharts – Formal Quotation and Budget Quote

Appendix D

Distributed Generation Initial Report

Appendix E

Second Iteration Prototype

Appendix F

Final Interface

Appendix G

Initial Project Plan