[Your project name] Requirements Specification

Version 1.0

27 February, 2017

Developed by:

Student Name ILENI TUDOR-ALEXANDRU

Group, year 931, 3

Version History

Version	Description of Change	Author	Date
V01	Initial/Modification of ….	ILENI TUDOR	27 February, 2017

Contents

1. Introduction

The application helps the electricity company to keep track the clients and invoices. This will help the company and clients to have a situation of the monthly payments.

Provide an overview of the system and some additional information to place the system in context.

1.1.Purpose

The purpose of the document is to present the functionalities and the characteristics of the software application **Electrica Admin** that aim to help the electricity company to keep track of its clients and monthly payments.

Provide an overall description of the Functional Requirement Document, its purpose. Reference the system name and identifying information about the system to be implemented.

1.2. Scope

The scope of the document is to explain how to system behave, what are its functionalities and how to implement them. The application context and the users are explained in the section 2 and the requirements are specified is section 3.

Discuss the scope of the document and how it accomplishes its purpose.

1.3. Definitions, Acronyms, and Abbreviations

Define terms, acronyms, and abbreviations used in the FRD.

1.4. Document Overview

The paper has 4 main parts. In the first part it presents an introduction regarding the purpose and the scope of the software and also the paper itself. In the second part is show the integration of the application in the context and also some user characteristics. The third part helps the programmers to understand the solution domain, and explains the requirements and also the input and the output of the system. The last part introduces the main use cases of the system.

Provide a description of the document organization.

2. Product/Service Description

The application must help the administer of the electricity company to keep track of all its clients and them invoices. He can add then and list the invoices. The specific requirements are presented in section 3.

In this section, describe the general factors that affect the product and its requirements. This section should contain background information, not state specific requirements (provide the reasons why certain specific requirements are later specified).

2.1.Product Context

The application in independent.

How does this product relate to other products? Is it independent and self-contained? Does it interface with a variety of related systems? Describe these relationships or use a diagram to show the major components of the larger system, interconnections, and external interfaces.

2.2.User Characteristics

The application will be used by the administrator of the electricity company. He is part of the staff and it supposed to have experience to work with computer, but also some technical expertise to work with large amount of data.

Create general customer profiles for each type of user who will be using the product. Profiles should include:

- *Student/faculty/staff/other*
- *experience*
- *technical expertise*
- *other general characteristics that may influence the product*

3. Requirements

REQUIREMENTS:

1. ADD A NEW CLIENT
    - The user must introduce the following information:
        - Name : String, maximum 256 characters
        - Address: String, containing the no. street, the street name, city
        - Id: String, unique id for each client
    - As an output the system will store the client in the database and print the corresponding message.
    - For saving the client the controller 'addClient' method is called and the repository 'add' method. The method from controller will produce the message based on the response from repository.
2. ADDING THE MONTHLY INDEX COUNTER
    - The user must introduce the following information:
        1. Name : String, maximum 256 characters
        2. Address: String, containing the no. street, the street name, city
        3. Id: String, unique id for each client
        4. Year: Integer YYYY
        5. Month: Integer MM
        6. The sum to pay: float representation
    - As an output the system will store the index in the database and print the corresponding message.
    - For saving the invoice, the controller '`AddClientIndex` method is called and the repository 'add' method. The method from controller will produce the message based on the response from repository.
3. LIST THE CURRENT INVOICES
    - The user must introduce the following information:
        1. Name : String, maximum 256 characters
        2. Address: String, containing the no. street, the street name, city
        3. Id: String, unique id for each client
    - As an output the system will print all the invoices related to the found client or nothing otherwise
    - To get the invoices the method ListIssues form controller is called, it gets all the issues and filter them accordingly

- *Describe all system requirements in enough detail for designers to design a system satisfying the requirements and testers to verify that the system satisfies requirements.*
- *Describe every input into the system, every output from the system, and every function performed by the system in response to an input or in support of an output. (Specify what functions are to be performed on what data to produce what results at what location for whom.)*
- *Each requirement should be numbered (or uniquely identifiable) and prioritized.*

3.1. Functional Requirements

List the functional requirements (FR) of the system.

Section/ Requirement ID	**Requirement Definition**
FR1.0.	The system shall be able to add a new client
FR1.1	The system shall be able to add a new invoice
FR1.1.1	The system shall be able to list the invoices for a specified client

3.2.User Interface Requirements

The home page may present menu from where to select a functionality: ADD CLIENT, ADD INVOICE, LIST ALL INVOICES.

1. When selecting ADD CLIENT a submenu must be printed and ask the user to provide the name, address and id of the client. In case of GUI a button SAVE must be present. When click it, a message box must show up with the corresponding message or error.
2. When selecting ADD INVOICES a submenu must be printed and ask the user to provide the name, address and id of the client and also the year, month and the sum to pay for the current invoice. In case of GUI a button SAVE must be present. When click it, a message box must show up with the corresponding message or error.
3. When selecting LIST ALL INVOICES a submenu must be printed and ask the user to insert the name, address and id of the client. The corresponding invoices will be printed, or 'no invoices' message in case that the user has no invoices. In case of GUI the invoices will be printed in a list box, or the message will be printed in a message box.

In addition to functions required, describe the characteristics of each interface between the product and its users (e.g., required screen formats/organization, report layouts, menu structures, error and other messages, or function keys).

3.3.Usability

Include any specific usability requirements, for example,

Learnability

- *The user documentation and help should be complete*
- *The help should be context sensitive and explain how to achieve common tasks*

- *The system should be easy to learn*

3.4.Data Management

The main entities used by the application are Client and Issue. Any Issue has a Client, so the relation between them is One To One.

A Client has the following fields (that are stored in the file):

1. Id: String, default value = “”, it may be formed of any character
2. Name: String, default value =”” it may be formed by digits and letter
3. Address: String, default value =”” it may be formed by digits and letter and comma

An Issue has the following fields (that are stored in the file):

1. Id: String, default value = “”, it may be formed of any character
2. Year: Integer, default value = 0, must have the format YYYY
3. Month: Integer, default value = 0, must have the format MM
4. ToPay: real number, default value = 0
5. Paid: real number, default value = 0

Specify the requirements for any information that is to be placed into the files, including

- *types of information used by various functions*
- *data entities and relationships*
- *valid range, accuracy, and/or tolerance*
- *units of measure*
- *data formats*
- *default or initial values*

4. User Scenarios/Use Cases

The main actor of the system is the ADMIN of the electricity company. The use cases are (they are explained in the ‘AnalysisDesignDocument’ at the paragraph 3):

(Add new client)

Description: The administrator insert data about the new client and then persist it

Precondition: The client does not exist, name and address are valid

Postcondition: The client is saved in the list of clients

(Add new invoice)

Actors: Admin

Description: The administrator insert data about the new client, invoice and then persist it

Precondition: The client exists, name and address are valid, YEAR, MONTH TOPAY are numbers

Postcondition: The new invoice is saved in the list of invoices

(List current invoices)

Actors: Admin

Description: The administrator insert data about the client and see its invoices

Precondition: The client exist, name and address are valid

Postcondition: The list of its invoices is printed or “No invoices”

Provide a summary of the major functions that the product will perform. Organize the functions to be understandable to the customer or a first time reader. Include use cases and business scenarios, or provide a link to a separate document (or documents). A business scenario:

- *Describes a significant business need*
- *Identifies, documents, and ranks the problem that is driving the scenario*
- *Describes the business and technical environment that will resolve the problem*
- *States the desired objectives*
- *Shows the “Actors” and where they fit in the business model*
- *Is specific, and measurable, and uses clear metrics for success*