UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CIÊNCIA DA COMPUTAÇÃO (COMPUTER SCIENCE)

MATHEUS PRIEBE BERTRAM

A Trust-Based Social Network Proposition for a Remote Homecare System

Diplomation Project.

Prof. Dr. Valter Roesler
Advisor

Porto Alegre, March 2012

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitora: Prof. Carlos Alexandre Netto
Pró-Reitora de Ensino: Profa. Valquiria Linck Bassani
Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb
Coordenador do curso: Prof. Dr. Raul Fernando Weber
Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

ABSTRACT

In the present work, we present a trust-based social network for a remote homecare system that has been implemented and validated. It consists of a Web platform where users can post contributions and interact among themselves by commenting and clicking the ”I like it” button on a given contribution. A group of 12 people took part in the testing of the work by actively participating on the social network over a period of 10 days. The main objective of this work is to introduce the concept of trust into this network and propose a way to sort information which is based both on timing and trust among users. Some interesting results have been achieved concerning the small-world social network structure of the resulting graph - namely the presence of communities, a high clustering coefficient and a low average path size. Furthermore, we introduce the concept of reputation inside the social network by applying the *PageRank* algorithm on the resulting trust graph and suggest the incorporation of this concept as a future work on the attribution and evolution of trust in social networks. As for the proper validation, we show that the present work proposes an information sorting tool that fits better to the ideal information sorting for general users than a timestamp-based sorting, which is used in several applications nowadays.

Keywords: Social Computing, Trust in social networks, Remote Homecare, Trust-based algorithms, Information sorting.

RESUMO

No presente trabalho, apresentamos uma rede social baseada em confiança para um sistema de homecare remoto que foi implementada e validada. Ela consiste de uma plataforma Web onde usuários podem postar contribuições e interagir entre eles comentando e clicando *no botão "Gostei"em uma dada contribuição. Um grupo de 12 pessoas tomou parte nos testes do trabalho participando ativamente da rede social por um período de 10 dias. O objetivo principal deste trabalho é introduzir o conceito de confiança nesta rede e propor uma maneira de classificar informações que é baseada tanto em* timing *quanto em confiança entre usuários. Alguns resultados interessantes foram alcançados com respeito à estrutura* small-world *de redes sociais no grafo resultante - nomeadamente a presença de comunidades, um alto coeficiente de clusterização e um tamanho médio de caminhamentos baixo. Ainda, introduzimos os conceito de reputação na rede social aplicando o algoritmo de* PageRank *no grafo de confiança resultante e sugerimos a incorporação desse conceito como um trabalho futuro na atribuição e evolução de confiança em redes sociais. Quanto à validação, mostramos que o presente trabalho propõe uma ferramenta de classificação de informações que se ajusta melhor à classificação ideal destas informações para usuários genéricos quando comparada à classificação baseada em* timestamp, *que é usada em várias aplicações atualmente.*

Palavras-chave: Computação Social, Confiança em Redes Sociais, Homecare Remoto, Algoritmos Baseados em Confiança, Classificação de Informações.

ACKNOWLEDGEMENTS

I would like to thank, first of all, my parents Edela Priebe Bertram and Paulo Bertram, as well as my brother Samuel Priebe Bertram, for helping me in every way through all my studies.

I wish to thank as well all the people who helped in the process of this work: Prof. Dr. Valter Roesler, for advising me; Fernando Varella, Francisco Krahe, Germano Caumo Carniel, Guilherme Lazzarotto Lima, Iulia Dolganova, Leonardo Crauss Daronco, Leonardo de Miranda Borba, Luis Augusto de Oliveira Brandelli, Maria Lúcia Kroeff Barbosa, Mateus Bisotto Nunes and Tobias Brignol Petry who helped with testing and revising the work.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

1 INTRODUCTION

The use of Information and Communication Technology (ICT) tools in health care, these days, has been much researched and worked upon. On this subject, the term *Telemedicine* was coined, and means, as defined by Perednia and Allen [33], ”the use of telecommunications technologies to provide medical information and services”. About the use of technology in medicine (or health care), Hanseth and Aanestad [21] say: ”telemedicine is expected to enable huge improvements of health care services through radically new and improved ways of collaboration and organizing within the health care sectors”. In the context of this work, the use of ICT in a homecare system (more specifically, that forms a remote homecare system) should bring several advantages to this activity. We discuss here the impacts of a remote health care system integrated with a social computing tool on the treatment of patients that receive treatment remotely on their homes, and use their television-sets to treat themselves, supervised by specialized doctors. Specifically, we focus on a part of the system, which is a social network inside the software that enables the interaction among the participating patients. The question we are looking forward to answer here is ”how can such a social network improve the health status of the system's users?”.

1.1 Homecare and Remote Homecare

Homecare, as described by Falcao [17], is a very wide concept, but usually means the assistance and monitoring of patients in their own home 24 hours a day by specialized nursery personnel.

That being, we can also define what we call *Remote Homecare* to be a homecare service where the monitoring can be made in great part with the use of a remote monitoring system - with the use of cameras and sensors that measure the health signals of the patients and send them to a remotely localized specialist.

The project dealt with within this work is all about that: monitoring patients in their homes with the use of technology to fetch vital signs from patients and assist them remotely.

1.2 Objective

The main objective of the present work is to show that patients of a remote homecare system can benefit from virtual social environments. The area of social computing [40], therefore is of primary importance for this task. This main objective may be subdivided in two different flows:

- the system should help the patients to interact among themselves in order to reduce some possible isolation brought by some disease being treated with use of the homecare system or even by the use of the system itself, which is done at home, and therefore may naturally isolate the users from the social living. On this subject, it is said by Jantsch et al. [22] that many get naturally isolated. As for older people, e.g., their children grow up and move to go on with their lives. After some years, they start living alone because their partners eventually pass away. Many times, the living one does not want to bother its children living with them, or there is no space in their home to do so, e.g., and they end up living alone or in a senior home. This isolation may eventually bring depression, and so the system wants to reduce such depression possibility through talks with friends, parents, and so on;

- the system should also help the patients to improve their health condition, in the sense of keeping - or, in the worst case, bringing back - the overall health status of some given patient to a controllable and positive equilibrium, in order to extend the longevity of this patient. The system could be used to obtain knowledge about some situation (e.g. a chronic disease) through interactions via a system of recommendations (videos, text, and so on) or a system of questions and answers, e.g;

In this context, a social environment that facilitates the treatment of patients of a remote homecare system and makes these patients health measures better and helps them to be kept in a state of equilibrium is of undoubted importance. Such social environment is yet to be explained in further sections - as well as which tools are used to achieve the above-mentioned goals.

1.3 Motivation

Every ICT tool that has the potential of helping improve the health care system of some region is interesting for the society and should therefore be studied.

As Lima [16] points out, some countries' public health care systems - e.g. Brazil's system - do not provide enough beds to every patient on hospitals. There is to this date a great lack of support from the public health care system to the population. In this sense, a tool that helps treat patients in their own home (homecare) can be useful not only to those patients who have difficulties in leaving their home for some reason, but also to help minimize the problem of this lack of support from the government.

Furthermore, the use of a collaborative system where users share interesting ideas on health care can be of great benefit to these users. Any user who has passed through some situation before - e.g. a diabetic patient who had problems eating mango - can transmit the knowledge quickly to many other users, who may benefit greatly from such information. Sometimes doctors may not have specific practical knowledge about a disease, and therefore would not be able to transmit useful information on that. An integrated system, where doctors can share interesting contributions as well as mediate some other contributions (since some patients may post erroneous information), would be the ideal here.

Yet to be mentioned is the influence of social contact in the treatment of patients. According to Jantsch et al. [22], older patients who are treated in their homes start suffering from reclusion, and that reclusion may lead to depression. A little deeper into this subject goes Uvnas-Moberg [38], who points out other physiological and endocrine effects of social contact. According to Uvnas-Moberg, when some living being has warm or, more generally, positive surroundings, such situation leads to storage of energy and growth, by affecting the anabolic metabolism. In opposite situations - with bad surroundings (e.g. fights) - the metabolic metabolism would be affected, leading to the use of calories for the performing of locomotor activities. That all means that good social surroundings lead to relaxation, whilst bad surroundings end up leading to the opposite of relaxation. The system proposed by the present work also visualizes social contact as something good and favourable to the treatment. Patients would communicate with others and share ideas, therefore reducing isolation and feeling ”relaxed” from the social contact, making the treatment possibly more effective.

1.4 Structure of this document

We start off after this brief introduction on the case by introducing some ground concepts for further understanding of the present work in the chapter **Basic Concepts**.

That done, we continue to pointing out the state of the art concerning social networks, social computing and trust in social computing in the chapter **Related work**.

Moreover, in chapter **Social Computing**, an overview on the term *Social Computing* is given as well as a report of its presence and importance in todays world and in the future. In the same section, the importance of social networks - here using the general term of social networks - in health care is presented.

The iCare Project, which is the basis for the development of the present work, is explained through in the chapter **The iCare Project and Communication Amongst iCare users**. All of the fundamental aspects of the project are clarified, going through motivation, composition and results of it. The composition of the social network proposition inside the iCare project is revised in this same chapter. There, an insight into the contribution system is given, as well as an explanation of the fundamentals of its functioning through trust - we go through the subject *trust in social networks* and expose in details how the trust network is built up inside the system.

In the **Development** chapter, insights into the implementation are given, going through every fundamental aspect of the development of the system - including database details, code structure details, details on the concept of trust as it is viewed within this work and so on.

In the chapter **Validation**, the testing and validation methodology is described in details, with regard to the testing environment which is detailed in **Annex A**.

All of the results of the validation phase of the social network are presented in the chapter **Results**, and we conclude over the results and other details of the project in chapter **Conclusion**.

In the chapter **Future work**, we intend to predict which can be relevant to work upon in the future - in the context of social networks, trust and health care.

1.5 Contribution of this work

The present work contributes in many aspects to the ongoing research on social computing, trust in social networks, homecare and health care in general.

To start off, it is important to say that the system presented in this work applies a functional trust-based system in a homecare network which is somewhat close to a real situation - the users are real people, but the application is more restrict in respect to a real homecare network. Both areas could benefit from the advances in this research: both social computing and homecare (or health care in general terms). The use of a collaborative system through contributions can provide patients in general with a better treatment and the research area of social computing may benefit from interesting interpersonal interactions.

Furthermore, it is interesting to create a unified system where both doctors and patients can contribute with valuable information. If every user has his ”trust-network”, i.e. people on the network to which he carries a higher trust, as well, it may be interesting to observe how much benefit in the form of good information that network would provide him. Doctors can naturally have a bigger trust, since they are the ones that can be trusted over all, but it is important to separate relevant information to the user from irrelevant information to the user. That can be done through a profile similarity function, which combined to trust and a defined time-span can deliver recent, trusted and relevant information to the user. With all that, this system can also provide a real comparison between reality and theory in social computing and homecare.

The usability question is also a very important one. From scratch it has been one of the goals of this work. To deliver relevant, timely and trusted information to the user without much need to wander through the network to find valuable information. It is important to refer that the system has been developed to fit in a television screen, but it would be possible to adapt it to mobile devices, for example.

2 BASIC CONCEPTS

Before we start off with the deeper explanation of the present work, it is important to focus on some ground concepts.

2.1 A brief conceptualization of trust

As said by Seligman, ”the existence of trust is an essential component of all enduring social relationships” [36]. Besides, according to Abdul-Rahman and Hailes [1], trust decisions are made, directly or indirectly, in most of our everyday lives - e.g. when purchasing goods in a shop, we tend to choose brands that we trust the most.

The concept of trust may have different meanings depending on the research context, situation which is primarily driven by the specific definition of trust depending on which study is being made [14]. As for a more concrete definition, we can say adaptively that trust can be defined in the following two actions [4]: first, the *trustor* gives *trustee* the right to take an action; then, the *trustee* takes an action which affects both *trustor* and *trustee*. That is easily mappable to the context of this work: a user trust another to some level when he, to such level, takes the risks of the actions of this second user or, more specifically, the assertions made by this second user. If some trust level has been put into the contribution, it can be virtually transported to the system through the interaction between the users.

Still as said by Abdul-Rahman and Hailes [1], social interaction these days span multiple boundaries - cultural, geographical, and so on - and virtual communities are very similar to real ones; that being, if trust is applied on the human interactions on real communities, it should also be applied onto the virtual communities, since these are to date operated by humans.

2.2 Social Computing

As put by Parameswaran and Whinston, ”Social Computing and online communities have ushered in a new era of the web, where information and communication technologies are facilitating organized human endeavor in fundamentally new ways” [32]. While through the end of the twentieth century IT was mostly centered in business computing, the twenty first century brought with it the paradigm change to social computing, the scope has shifted from corporations to social organizations [32].

As defined by Wang et al. [40], *social computing* consists of the ”computational facilitation of social studies and human social dynamics as well as the design and use of ICT technologies that consider social context”, and is a research and application field that includes both computation and social sciences. Still according to Wang et al., the idea of social computing can be traced back to the decade of the 1940s, when a paper from Vanevar Bush entitled "As We May Think" [13] was published. In the paper, the author presented some ideas that would be thought about and researched in the future. But only in the 1960s, ARPA (Advanced Research Projects Agency), which would in the future lead to the Arpanet, predecessor of the Internet, was created.

In its early ages, social computing had two different *foci*, as Wang et al. say [40]. One of them was built around technological issues, interfaces, user acceptance, and social efforts for collaboration and communication. The other focus was on the computation techniques to facilitate social studies, which were specially used by political entities or general organizations, for example. In the past years, the scope of social computing has widened itself a lot, embracing many research and development areas.

Still said by Wang et al. [40], in order to support social interactions, social computing relies on "communication; human-computer interaction; sociological, psychological, economic and anthropological theories; and social network analysis". Moreover, both ICT and society benefit from each other – on one hand, social computing has emphasized technological development for society and, on the other hand, ICT development has incorporated social theories and practices. Wang et al. [40] still state that the public sector is an important application area of social computing – and that includes healthcare. Other application areas vary from web tools (such as blogs) to forecasting tools (simulations applied for different areas, e.g. market analysis). In the public health domain, social computing can provide a system design methodology and guidelines for specific system functions and how people and the community interact with the system and among themselves.

The authors [40] still strengthen the forecast that social computing will move into social intelligence, a state beyond information processing. That may bring several benefits for all the areas reached by social computing, including, as already mentioned above, healthcare.

Social computing promises to be very influent in many areas such as business and politics. As a matter of fact, computing itself is shifting to a cloud-oriented paradigm, and even operating systems are gradually turning from personal to network-centric; furthermore, the potential of social computing may offer opportunities for both researchers and business men [31].

2.2.1 Importance of Social Networks in Health

The influence of social networks - the wider concept of social networks, not yet applying it to Computer Science - was first proved in the 1970s [37], with empirical results showing that social networking could affect mortality rates. Still pointed out by Smith and Christakis [37], social networks affect health through several mechanisms, such as (all according to Smith and Christakis [5]):

- provision of social support - emotional impulses, appraisal, e.g.;
- social influence - e.g. norms, social control;
- social engagement - social roles, interpersonal attachments;
- person-to-person contact - physical contact in a first instance, but mappable to virtual social networks;

- access to resources - such as money or information, e.g.;

All of the above-mentioned mechanisms are trivially mappable to virtual social networks as follows, considering the system being presented in this work:

- provision of social support - in a contribution system, users may give positive feedback to contributions from others, which has an emotional/sensorial impact;
- social influence - users under treatment should be open to their ”superiors”. In this sense, a contribution given by a doctor, for example, should be taken as a guideline, and the adherence of the user to such contribution could make its health status improve;
- social engagement - a good contribution of any kind is favourable to health status improvement;
- person-to-person contact - this may be tightly liked to groups of common users, relations between them and trust in the social network;
- access to resources - information is the key concept here. In a contribution system, information is all around for the seizing by everyone, and is the most valuable good in the system. A given hint by a diabetic user, for example, on the effects of a given food on his health status, can be very useful to any other diabetic user;

As Smith and Christakis [37] say, *social support* studies are able to demonstrate that socially isolated individuals are less capable of buffering health stressors and are, therefore, more susceptible to negative health outcomes. In the sense of *social support*, helpfulness is all about how the people related to some individual are able to help such individual through informational support, appraisal support or emotional support.

2.3 Influence and epidemics in social networks

Social networks play an important role on dynamics of epidemics [28] and influence [23]. According to Moore and Newman [28], ”the structure of social networks plays an important role in the dynamics of disease propagation”, and small-world networks (concept further defined) show a much faster disease propagation than regular networks.

In a study by Christakis and Fowler [15], the spreading of obesity in a large social network was observed in the course of 32 years. A dense network of 12067 people was evaluated and an increase in the obesity levels over this time has been observed.

The intention of the work was to observe the person-to-person spread of obesity, and the results were the following [15]:

- Discernible clusters - which extended up to three degrees of separation - of obese people were present in the network, suggesting that close interpersonal relations determined in the obesity increase - being the graph locally dense;
- These clusters apparently were not formed by obese people that started to relate with each other. They were created because there was an increase in the probability of one becoming obese in case one of its acquaintances should become obese;
- There were high increases in the probability of one becoming obese in case one of the following should first become obese:

- Adult sibling to the person in question;
- Spouse of the person in question;
- Friend of the person in question.

For a conclusion, Christakis and Fowler [15] took that network effects seemed relevant to the development of obesity.

That can be also applied to the trust network of the present work. It is interesting to observe the formation of cycles of trust, where people who have common trusted people tend to trust each other. This way, it is important to apply a certain level of transitivity to trust, since people with acquaintances in common tend to trust each other to a certain level.

2.4 Connectivity between agents and its relation with trust

There is a very interesting work by Steve Milgram [27], which shows that the average size of paths between two different people in an acquaintance network is 6. This characteristic of social networks has been called *small-world*.

In this work, Milgram starts of by formulating the small-world problem as the probability of two different people - say A and Z - in an acquaintance network knowing each other. He then expands the formulation to the number of people in a path in this network between the first two people - A and Z [27].

The experiment process can be summarized by **Algorithm 1**. To explain it briefly, a message M has to be delivered from some randomly chosen citizen of the USA (United States of America) A to some other randomly chosen citizen of the USA Z, which is different from A.

In case A knows Z, he delivers M directly to Z. In case there is no direct acquaintance relation from A to Z, A chooses an acquaintance which is most likely to know Z and delivers the message to it. This process is repeated until Z receives the message.

Algorithm 1 Milgram Experiment

Let U be the set of all people for the experiment
$A \leftarrow$ choose random element in U
$Z \leftarrow$ choose random person in $(U - \{A\})$
Let M be the message to be delivered from A to Z
while $\neg receiveMessage(Z, M)$ **do**
    if A knows Z **then**
        A delivers message to Z
    else
        $A \leftarrow$ intermediary person most likely to know Z
        A receives message M
    end if
end while

From the 160 chains - i.e. paths to deliver the message M - that were started, 44 were finished. Out of these finished chains, there was a variation of 2 to 10 intermediary people, with a median of 5. As a result concerning the small-world situation, two random people were connected by a path of six acquaintances [27] i.e. *a degree of separation* of six.

This small world situation is exactly what happens in social networks, being them virtual or real.

In a recent study, it has been shown that the average distance between users in Facebook - one of the broadly used social networks in the current time - is in average $4,74$ [2]. That means that the degree of separation is $3,74$, which can be rounded up to 4. Since Facebook is a social network that is worldwide spread, with circa 721 million users, we have a good sample of the acquaintance relations in the whole world, and that is why this is a considerable result.

It is of interest to observe the average shortest path and community formations - still to be explained - in the resulting trust network of the present work. The result will most likely be a small-world situation, which is a very strong characteristic of social networks in general. Since the present work has been tested with real human behaviour - the testing and validation methodology is still to be explained - and a trust network is intuitively a social network, we will most likely have a small-world situation.

2.5 Trust in social networks

To make the system more usable and more comfortable to the user, it is important to sort all the contributions for their importance. This can be done through a trust system inside the social network. Once a person "trusts" some other person, this other persons information is theoretically also trusted.

The concept of trust may be differently defined, but in this work we consider it to be reflexive,not symmetric and transitive, similarly to what is defined by Walter et al. [39]. That means that if, considering a set of agents A, an agent $\alpha \in A$ trusts an agent $\beta \in A$, β does not necessarily trust α. It is important to point out that we are dealing with trust in a quantified way. Similarly to Walter et al. [39], we define trust (t) between two agents to be such that $0 \leq t \leq 1$. Then, if we define the relation *trusts* to express the trust between agents, for $\alpha, \beta \in A$, what we can say concerning symmetry is:

$$\alpha\ trusts\ \beta \nRightarrow \beta\ trusts\ \alpha$$

As for the reflexivity of the relation, we define that every agent trusts itself completely - with $t_{\sigma,\theta}$ denoting the trust from a generic agent $\sigma \in A$ to another agent $\theta \in A$:

$$\forall \alpha \in A, (\alpha\ trusts\ \alpha) \wedge (t_{\alpha,\alpha} = 1)$$

Besides, we consider trust to be transitive, that is, if $\alpha \in A$ trusts $\beta \in A$ and $\beta \in A$ trusts $\gamma \in A$, than α trusts γ to some level (yet to be defined). That is,

$$\forall \alpha, \beta, \gamma \in A : (\alpha\ trusts\ \beta) \wedge (\beta\ trusts\ \gamma) \Rightarrow \alpha\ trusts\ \gamma$$

Still concerning the transitivity of the trust relation, if we have a transitive relation among agents α, β, γ such as described in the proposition above, the resulting trust from α to γ is defined as follows:

$$t_{\alpha,\gamma} = \delta(t_{\alpha,\beta} \cdot t_{\beta,\gamma})$$

In the equation above, we introduce one more coefficient (δ), which is the *attenuation coefficient* to the trust relation. As for this coefficient, we define it as follows - assuming that we want to know the transitive trust from α to γ:

$$\delta = \frac{1}{|B| + 1},$$

$$(B \subseteq A)|(\alpha\ trusts\ \beta_1) \wedge \left[\bigwedge_{i=1}^{|B|-1} (\beta_i \in B)\ trusts\ (\beta_{i+1} \in B) \right] \wedge (\beta_n\ trusts\ \gamma)$$

As the concept of trust tends to fade away when the trusting path becomes too large - intuitively, one does not trust another who is transitively too distant - we always look for the shortest path between trustor and trustee. This way, we want $B \subseteq A$ so that:

$$(\nexists B_1 \subseteq A)|B_1| < |B|$$

As for the transitivity question, many applications assume that trust is not transitive [39]. In the present work, we assume transitivity to a certain level, yet to be explained in the **Development** section.

2.5.1 The trust network

We can see the trust net in the contribution system as a directed graph [9]. The weight of an edge between two nodes on the graph illustrate the trust from an agent to the other - each denoted by a node. This is better explained in **Figure 2.1**.

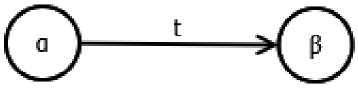

Figure 2.1: Agent α trusts agent β to some degree t

If there is reciprocal trust between to agents, it would be represented as explained in **Figure 2.2**, with edges in both ways between these agents.

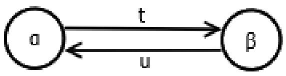

Figure 2.2: α trusts agent β to some degree t; β trusts α to some degree u

In the trust graph, we can build several trust relations between many pairs of agents. Moreover, there may be many paths - here, we consider paths where the number of edges is greater or equal to two, since a path with one edge is simply the direct trust from an agent to another - in this graph between pairs of agents (or nodes), and we have to decide which path should be taken to measure the final trust value - should simply the greatest of all values produced for the trust between two apart nodes (resulted from the multiplication of all of the weights of the edges on the path) be the chosen trust? This is yet to be answered in the **Development** section.

When talking about transitivity in the graph, we can assume as the *attenuation coefficient* simply the size of the path between trustor and trustee.

In **Figure 2.3** we have an example trust graph. There, we have that:

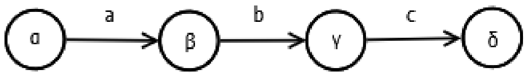

Figure 2.3: Trust graph example

$$t_{\alpha,\beta} = a$$
$$t_{\beta,\gamma} = b$$
$$t_{\gamma,\delta} = c$$

Let us now consider α as a trustor. Then, the trust values from α to the other nodes on the graph - given the aforementioned definitions:

$$t_{\alpha,\gamma} = \frac{1}{2} \cdot ab$$
$$t_{\alpha,\delta} = \frac{1}{3} \cdot abc$$

Still concerning transitivity, as already explained above, we consider in this work that we must always choose the shortest path when determining the trust from some trustor agent α to some trustee agent β. This is better explained in **Figure 2.4**. In the figure, we omit the trust values on both paths, as they are irrelevant in this case. The most important is that the shortest path between the aforementioned agents is chosen every time alternative paths exist. In this case, the shortest path between α and β is through ϵ. This way, we calculate the trust from α to β as:

$$t_{\alpha,\beta} = \frac{1}{2}(t_{\alpha,\epsilon} \cdot t_{\epsilon,\beta})$$

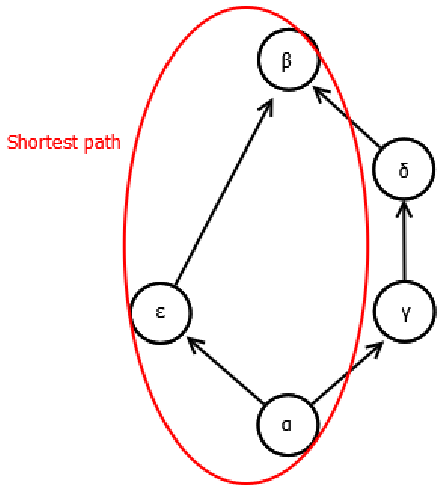

Figure 2.4: Trust shortest path

Furthermore on transitivity, should some hypothetical person trust some other hypothetical person when between them there is a path of many trust relations? Most likely, these two people do not even know each other. The level of deepness in the attribution of trust must be evaluated. This will also be explained in the following sections.

3 RELATED WORK

When talking about trust, its relation with computer science has been, in the past years, mostly applied to security - on the classic scenario where Alice needs to communicate with Bob through some channel when nobody else should understand [39].

As for research on the area of social computing and trust in social networks, we can refer the work of Walter et al. [39], for instance. In this work, an interesting recommendation model is proposed, where agents ask for recommendations and make decisions based on trust over a set of recommendations from other agents. This work has shown some considerable results. To begin with, the trust networks formed on the performed tests tended to converge to an optimum. That is, without explicit coordination, the agents ended up organizing themselves and long trust paths were formed, making it possible for some agent to trust another similar one who is localized very far in the network [39].

Abdul-Rahman and Hailes [1] propose in their work a discrete model of trust and reputation. They start off by defining trust as the subjective probability with which an agent will perform a given action both before it can monitor this action and in a context in which it will affect its own action [18]. Moreover, they assume that mathematical probability has properties that make it unsuitable for trust metrics, and assume a subjective trust model with different levels of trust [1]. Four degrees of trust are defined: from *very trustworthy* to *very untrustworthy*, passing through *trustworthy* and *untrustworthy*. As already mentioned, a reputation model is put to use as well. That is, the past experience counts when making a decision based on some agent, which ranges, also discretely and analogously to the trust model, from *very good* to *very bad*.

Mui et al. [29] propose a computational model of trust and reputation applied to e-business. In their work, each single agent belongs to a set of agents also known as an embedded social network. In addition, every action of trust, reciprocity and reputation can affect the overall status of this network. They build a schema to clarify this notion as cycle:

- if an agent increases its reputation in the embedded social network, the trust of the other agents in the network in the single agent in question should also increase;
- if an agent increases its trust on another, the first will reciprocate positively to the second with a higher likelihood;
- if an agent increases his reciprocating actions in the social network, its reputation in it should also increase.

Marsh [25] deals with trust, distrust and the absence of trust by defining trust as a value t where $-1 \leq t < 1$. This definition has shown, according to Marsh, to be not appropriate alongside with the algebraic operations defined in his work [25].

Furthermore, Marsh [25] points out that there is still work to be done on trust. According to him, trust is a subjective phenomenon, and humans tend to relate it with emotions, needs and so on. Then, in this aspect, agents used in simulations should reflect better what a human being acts like. That is, according to Marsh, ”not as impractical as it may appear” [25].

We propose to contribute to the research field in building computational models for the concept of trust. In this sense, this work has a base on the positive affirmations of the current research and introduces models for trust-based applications.

4 THE ICARE PROJECT AND COMMUNICATION AMONGST ICARE USERS

The iCare project is, as put by Jantsch et al. [22], a remote homecare system for elderly people.

An iCare user is a patient whose monitoring is done by distance by an specialized doctor or nurse. The patient stays at home, while the health authority may be geographically far from him.

The project is suitable for several devices, including television sets, computers and general mobile devices [3]. Moreover, it is developed at I9Access, a company situated in the Federal University of Rio Grande do Sul.

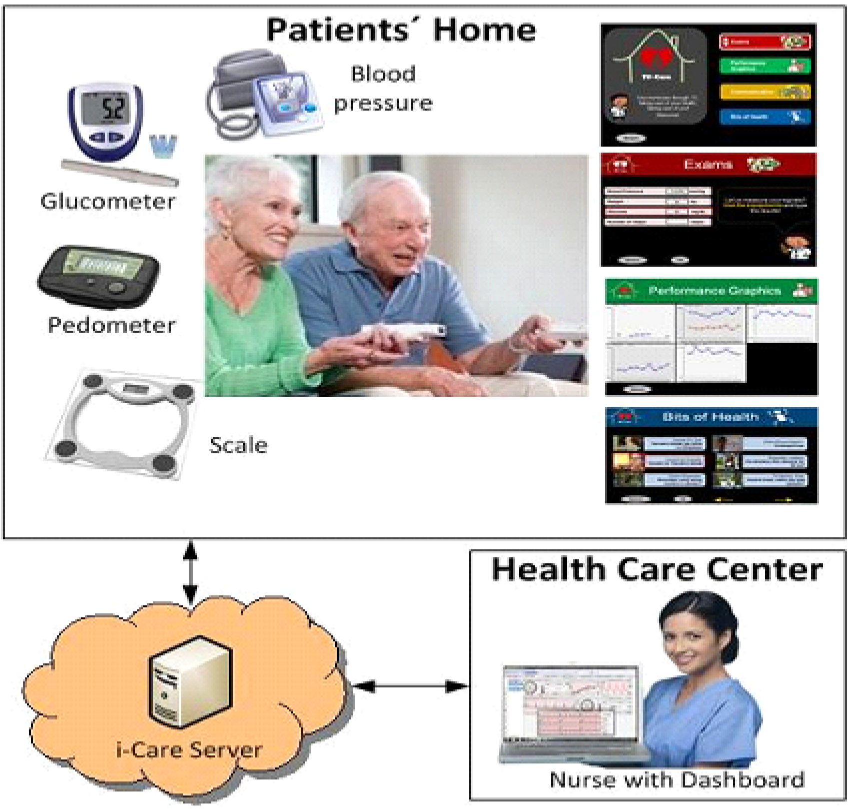

Figure 4.1: iCare system functioning

4.1 Motivation for the project

As Jantsch et al. point out, there is an increase in the number of aging people in the world these days, and a homecare service for these people can be of great usage. The system makes use of a television device to provide the interface with the user, observing the fact that most people of the target age have this device at home and are familiar with it. Based on a study from the WHO (World Health Organization), Jantsch et al. [22] still point out a great motivation for the work: there is an estimate that the aging people population will be greater than the children population in the year 2025.

4.2 What the project is made of

The project aims to provide autonomy to active elders in the following manner [3]:

- monitoring vital signs;
- being a tool for awareness of the users when dealing with their health;
- providing social interaction among users and, e.g., their families and friends - not yet developed in the time of the survey.

There are, basically, three main functionalities brought up by Jantsch et al. [22]:

- Realization of health measures and exams: the user, with some certain health profile, has to realize some required exams in order to measure how its health status is and what must be done to improve it. The software component concerning this point can be seen in **Figure 4.3**;
- Graphical display of the results from the above-mentioned exams: graphics show the measures taken above with the course of time, in order to better evaluate advances or setbacks. This is shown in **Figure 4.4**;
- A set of videos that provide health tips for the users: the users may watch different videos with several *foci* on different areas of health, such as diabetes, for example, and by doing so, they can also learn some tips on what to do to live a healthier life. This can be seen in **Figure 4.5**;

4.3 Results of the project

The iCare Project is still ongoing, but it has shown some important results this far, in terms of the users response to the interface. The first validation was done with 14 people. As shown by Jantsch et al. [22], circa 79% of the users have classified the interface as of easily readable and said the items on it were easy to be found.

Besides, the system has had a very positive evaluation from the users in terms of ”instructions to the user”: a hundred percent of the 14 users classified it as very good [22], and the usability has approved independently of the user profile.

Concerning effective health results on patients, the project has also shown these good results, for a group of three people [3]:

- the daily number of steps of the users increased, as they became more aware of their health condition and how they could improve it;

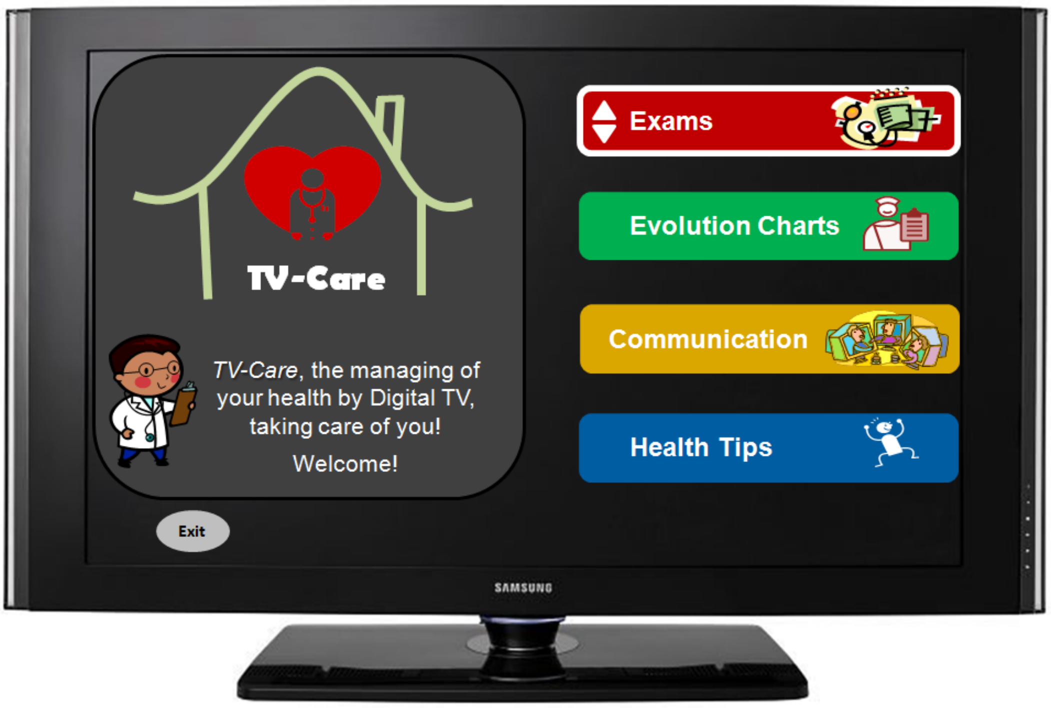

Figure 4.2: iCare system starting screen

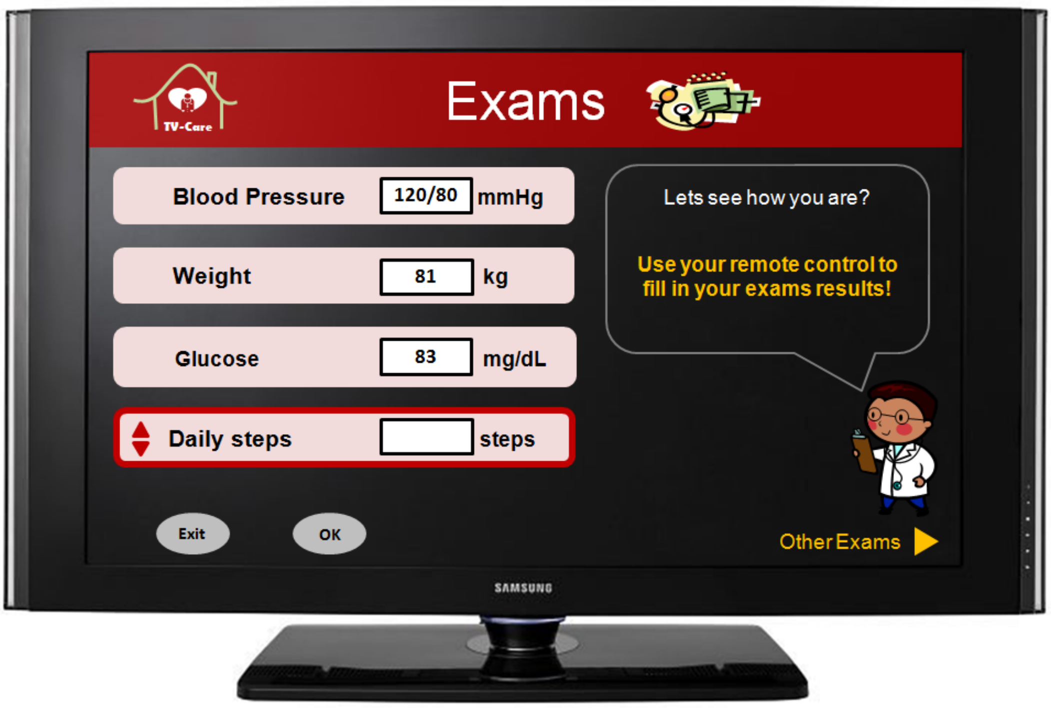

Figure 4.3: Exams screen on iCare

Figure 4.4: Performance charts screen on iCare

Figure 4.5: Health tips screen on iCare

- one of the three patients, who had a high glucose level in the beginning of the program, decreased his level to a normal after the period of testing;
- All the 3 patients on the experiment ended up losing weight, and some of them reduced their body fat by losing heavy weight and gaining lean weight;
- a diabetic patient managed to reduce his glucose levels in approximately 20%, a reduction that was very satisfying to him;
- even though not all of them lowered their main corporal indexes to the recommended level, there were satisfactory results on this matter, since all of them reduced their levels at least a little.

The iCare project is still being perfected, but, as we can see, it has already shown its applicability to a real situation.

4.4 Incrementing iCare

Despite the positive results, the iCare project still lacks on a tool for communication among users. This is exactly where the present work has its contribution, as here is developed:

- a social network, for the users to communicate amongst them and with the health authorities behind the system (doctors, nurses, and so on): still as researched by Jantsch et al. [22], there is a tendency among elderly people to live in reclusion. This tool aims at both approximating these people and providing a sharing platform to exchange ideas and improve the health status of every user of the system;
- the proposed social network must be usable and provide fast and easily acquirable information. That is why the social network developed in this work has a trust system to classify messages according to each users personality.

4.5 Communication amongst iCare users

The main goal of this work is to present and validate a communication tool for iCare users. Through such tool they are able to communicate themselves and share ideas and opinions that may be valuable to their treatment in some way. That being, the iCare user may connect itself to the proposed social network through the *Communication* channel and interact with other people in the system. Concerning this, the access of the user to important information should be also simplified. The user has access to the Communication channel on the main screen of the system, presented in **Figure 4.2**, where there are four options to choose, which were already explained above along with the whole explanation of the iCare project.

4.5.1 Contribution System

The contribution system works in such fashion, that there are several pages with room for contributions from users. When some user has certain knowledge which he considers worth sharing, he simply uses this room to make the other system users aware of the information. This is illustrated in **Figure 4.7**. It is important to refer that the interface has been developed with the idea that the screen should be easily viewed in a television set in mind.

The contribution system has the following elements:

- contribution input: this is where the user logged in the system can post his own ideas - or contributions in general. It is composed by a blank space for text input and a button of confirmation through which the user effectively inputs the contribution;
- contributions from other users: right below the contribution input stay the contributions from the other users. They are composed of:
  - name of the contributor - name of the user to whom the contribution belongs;
  - time of the contribution - time when the contribution was posted by the user in question;
  - the contribution itself - the data the user wants to share;
  - ”I like it” button - expresses if the current user has found the contribution interesting;
  - ”Comment” button - channel through which information related to the post in question can be added.
- Buttons ”Next” and ”Previous” to navigate through newer and older contributions respectively;
- ”Quit” button, to go back to the main screen of iCare.

The ”I like it” and ”Comment” functionalities have a big importance for this work. They are the elements which increase trust from one user to another - in a way still to be described.

When the user clicks on ”Comment”, he is directed to another page, very similar to the general contribution page, where the contribution to be commented is shown on top and the comments on it stay below - **Figure 4.8**.

Figure 4.6: ”I like it” and comment buttons - elements which build trust

4.5.1.1 Usability of the contribution system

As we can see in **Figure 4.7**, the system was developed so that all information could fit in one screen. Moreover, the information to be shown in each screen is classified based on *timing* and it actual importance to the user - this is done through the concept of trust in social computing, which will be explained further.

As it should fit in the system proposed by Jantsch et al [22], and the whole iCare system is developed with the objective that the system should be easily visualized in a television screen [22], the background is painted in black.

Figure 4.7: The contribution system

Figure 4.8: The comments for the respective contribution

Furthermore, all information present on the screen is printed in a good size, so that it may be visualized by distance. The navigation system that has been planned also takes the user from one contribution to another, until the end of the page, where the user can choose among going to the next page, going to the previous page or quitting the system (all may be seen in **Figure 4.7**).

The one problem here lies on the incapability of providing easily usable commenting (or, more generally, writing text) by the television set. That can be worked around, though, through the use of the ”I like it” button, which is very similar to the *Like* button in *Facebook* (http://www.facebook.com), for example. This way, the user may show its interest in the contribution with just one ”click”.

4.6 Conclusion

The iCare system has potential to be an important solution in remote homecare. The content of this work is inserted into this system, and intends to improve the health care of homecare patients through the efficient sharing of information.

5 DEVELOPMENT

In this section, the process of development of the present work is detailed. To begin with, we shall acknowledge the tools used to do so, and further specific implementation details will be provided.

5.1 Requisite Analysis

Details on how the system should be developed - tools, frameworks and other general project decisions - are presented below.

5.1.1 Implementation tools

For matters of implementation, the computational tools used are *Java* and *JavaServer Faces*. The integration of *OpenSocial* in the project might be of great use - for reasons to be explained below - and therefore is suggested as future work.

5.1.1.1 Java and JavaServer Faces

The iCare Project [22] uses Java and JavaServer Faces in its implementation. As the proposed social network is to be built upon this system, it is reasonable to choose compatible tools to do so. Besides, these are two broadly spread standards, well known by the ICT community and has plenty of documentation.

5.1.1.2 OpenSocial

According to Grewe [20], ”OpenSocial is a set of common application programming interfaces for Web-based social network applications”. The main reasons why such must be chosen to the continuity of the development of this work are:

- it is a broadly spread tool, supported by most of the state-of-the-art social networks;
- it has a solid documentation;
- it has a continuous growth and development, as it is a piece of open software;
- there is a great effort from the community behind it to turn it into a standard;
- finally, and most importantly, it provides the basic tools which are needed to the development of the proposed social network (people information, relationship information and activity information).

5.1.2 Integration with *Twitter*

According to Phelan et al. [34], social networks have become prevalent in the age of the Web, and the number of people who use the online social networks - such as *Facebook* and *Twitter* - has much increased in current times. A research has been done with promising results by Phelan et al. [34], demonstrating that *Twitter* can be a good channel for information recommendations.

In this sense, it is important if the current system is integrated with *Twitter*, for example. As it is a platform directed to the spreading of short messages, has many users, and has a good potential for information recommendations [34], any interesting contribution on *Twitter* concerning the issues common to the iCare system should be fetched by our contribution framework and shown to the users.

This point can be reinforced by the following example: let us suppose there is a doctor who has a new insight on a health matter and shares it on *Twitter*. The iCare users may find that contribution useful, and thats why such contribution should be provided in the iCare contribution system as well.

To fetch useful information from *Twitter*, the library *twitter4j* (http://www.twitter4j.org) is used and the useful information is gathered with use of queries on the social networks system. In this sense, every contribution with the content *tvcare* put on *Twitter* are fetched into the iCare contribution system.

The overall functioning of the algorithm is explained by the *pseudocode* in **Algorithm 2**, where ***startDate*** is the date from which the contributions should be fetched:

Algorithm 2 fetchTwitterContributions()

$twitter \leftarrow getTwitterInstance();$
$query \leftarrow "tvCare", startDate;$
$tweets \leftarrow findTweets(twitter, query);$

for $t \in tweets$ **do**
    $insertInContributionsList(t);$
end for

5.1.3 Tools

The main computational tool used in this work is, as mentioned before, *JavaServer Faces*. The system, then, consists of a web page with content backed up by controllers for such content, altogether with a database - making use of *Postgres* - system to store the contributions made for the user and other aspects that are still to be explained.

5.2 Implementation details

We should take a look at the specific details of the implementation. First, all of the database content will be explained.

5.2.1 Database details

There are 3 main tables developed to the context of this work. They are the following:

- *mbswallcontribution*: this table stores the contributions of all users on the system and is composed of the following fields:

- *cdlwallcontribution*: the primary key for the table, provides a unique identification for every contribution;
- *desname*: the name of the contributor, stored for fast access;
- *dessentence*: the contribution itself, i.e., the text input by the user;
- *islocal*: as the system is thought to be integrated with worldwide spread social networks, this is the field that expresses the locality of the contribution;
- *contimestamp*: a timestamp for the contribution, generated when it is saved on the database;
- *fathercdlcontribution*: a foreign key to the same table in question (mbswallcontribution), represents if the contribution in question is a contribution or a comment to a contribution - inputs with no father are normal contributions, while the ones with a certain *fathercdlcontribution* are a comment related to another contribution;
- *desusercodusuario*: a foreign key to the user which made the contribution, points to a table of users of the system.

	cdlwallcontribution [PK] bigint	desname character varying(255)	dessentence character varying(255)	islocal boolean	contimestamp timestamp without time zone	father_cdlwallcontribution bigint	desuser_codusuario character varying(255)
1	34	I9Access Admin	Teste i9access #1.	TRUE	2012-01-04 10:38:23.008		i9access
2	35	I9Access Admin	Teste i9access #2.	TRUE	2012-01-04 10:38:33.835		i9access
3	36	I9Access Admin	Teste i9access #3.	TRUE	2012-01-04 10:38:43.912		i9access
4	37	I9Access Admin	Teste i9access #4.	TRUE	2012-01-04 10:38:55.643		i9access
5	38	Matheus Bertram	Teste mpbertram #1.	TRUE	2012-01-04 10:39:22.098		mpbertram
6	39	Matheus Bertram	Teste mpbertram #2.	TRUE	2012-01-04 10:39:33.648		mpbertram
7	40	Matheus Bertram	Teste mpbertram #3.	TRUE	2012-01-04 10:39:44.752		mpbertram
8	41	Matheus Bertram	Teste mpbertram #4.	TRUE	2012-01-04 10:39:57.401		mpbertram
9	42	Arnaldo Teste	Teste Arnaldo #1.	TRUE	2012-01-04 10:40:27.028		arnaldoteste
10	43	Arnaldo Teste	Teste Arnaldo #2.	TRUE	2012-01-04 10:40:34.461		arnaldoteste
11	44	Arnaldo Teste	Teste Arnaldo #3.	TRUE	2012-01-04 10:40:40.555		arnaldoteste
12	45	Arnaldo Teste	Teste Arnaldo #4.	TRUE	2012-01-04 10:40:47.561		arnaldoteste
13	46	Pessoa Teste	Teste Pessoa #1.	TRUE	2012-01-04 10:41:14.406		pessoateste
14	47	Pessoa Teste	Teste Pessoa #2.	TRUE	2012-01-04 10:41:23.884		pessoateste
15	48	Pessoa Teste	Teste Pessoa #3.	TRUE	2012-01-04 10:41:31.83		pessoateste
16	49	Pessoa Teste	Teste Pessoa #4.	TRUE	2012-01-04 10:41:39.835		pessoateste
17	50	I9Access Admin	Teste i9 #5.	TRUE	2012-01-05 15:52:57.071		i9access

Figure 5.1: *mbswallcontribution*, the table of contributions

- *mbswallcontributionassoc*: this table expresses the people who are interested in a given contribution, and is composed of the following fields relating the tables *mbswallcontribution* and a table of users:
  - *mbswallcontributioncdlwallcontribution*: a foreign key to the table of contributions (*mbswallcontribution*), representing the contribution itself;
  - *interesteduseradmusuario*: a foreign key to the table of users, indicating that the referenced user is interested in the given contribution;

- *admusuariotrust*: this table contains information about the trust between pairs of users. It is important to remember that trust is not symmetric, at least in the context of the present work. Then, the fields from the table are the following:
  - *cdlusuariotrust*: primary key for the table;
  - *srcusuario*: the user who trusts;
  - *dstusuario*: the user who is trusted;
  - *trust*: the trust value itself between the users.

	mbswallcontribution_cdlwallcontribution [PK] bigint	interestedusers_codusuario [PK] character varying(255)
1	41	i9access
2	45	i9access
3	46	mpbertram
4	47	mpbertram
5	47	pessoateste
6	48	i9access
7	48	mpbertram
8	49	i9access
9	49	mpbertram
10	50	i9access
11	50	pessoateste
12	63	i9access
*		

Figure 5.2: *mbswallcontributionassoc*, the table of users interested for some contribution

	cdlusuariotrust [PK] bigint	trust double precision	dstusuario character varying(255)	srcusuario character varying(255)
1	4	0.392	pessoateste	i9access
2	5	0.554300016125799	i9access	i9access
3	6	0.891926625423278	pessoateste	mpbertram
4	7	0.5	i9access	pessoateste
5	8	0.5	pessoateste	pessoateste
6	9	0.5	arnaldoteste	i9access
7	10	0.52318831191153	mpbertram	i9access

Figure 5.3: *admusuariotrust*, the table of trust between a pair of users

5.2.2 Global code structures

The code is structured with the following elements:

- Webpage: the user interface, outputs to the user and takes input from the user to be dealt with by the controller;
- Controller class: backs up the web-page, providing it with data and general functioning;
- Entity class: a class that represents a database structure - e.g. a set of tables;
- Service class: accessed by the Controller class, does operations on the database concerning the tables related to it.

The interaction between the above-mentioned classes is characterized in the following manner: the webpage is backed up by the controller class, which provides it with content. This controller usually contains an Entity or a list of Entities on which to work upon and also gets information from the database through the Service class.

5.2.3 The controller class

The controller class is the heart of the system's functioning. It fetches information from the database, controls the use of entity classes (equivalent to database tables), demands recordings into the database, controls the webpage displayed to the user, does calculations on trust and takes care of all the contributions.

Besides providing the webpage (interface with user) with data, the controller class handles the contributions and the trust from the current user to every other. In this matter, it contains a mapping of unique contribution identifiers to the trust the current user has on the provider of that contribution. The mapping m may be defined as follows:

- $A \subset \mathbb{Z}$ is the set of unique identifiers for the contributions in the system;
- $B \subset \Re$ is the trust that the current user has on the given contribution, related to the trust that the current user has on the contributor in question - where $B = \{b \mid 0 \leq b \leq 1\}$;
- $m : A \longrightarrow B$ is the mapping of each contribution to the trust the user has on it.

The contributions will be displayed according to this mapping, with the addition of a temporal aspect - the contributions are sorted in subsets according to their *timestamp* - present on the table of contributions (*mbswallcontribution*). Each of these subsets are then sorted according to the mapping. Let us suppose that the algorithm is called A, and receives all the contributions as a parameter. Then, it works in the fashion of **Algorithm 3** - where C is the whole set of contributions and s is the size of each subset of contributions.

In order to explain **Algorithm 3**, we have to explain some points.

First of all, we have to define the relations $\leq_{ts}$ and $\leq_{tr}$, as well as the relations $<_{ts}$ and $<_{tr}$, as the relations for *timestamp less or equal*, *trust less or equal*, *timestamp less* and *trust less* - where *Timestamp* is the set of all timestamps and *Trust* is the set of all values of trust:

- $\leq_{ts} \subseteq Timestamp \times Timestamp,\ t_1 \leq_{ts} t_2 \Leftrightarrow t_1$ comes before or at the same time than t_2;

Algorithm 3 A(C, m)

$(X, \leq_{ts}) \leftarrow sortTimestamp(C)$;
$Y \leftarrow \{\emptyset\}$;
for $i = 1$ to $\lceil |X|/s \rceil$ **do**
    $x_1 \leftarrow minimal((X, \leq_{ts}))$;
    $X_1 \leftarrow \{x_1, x_2, ..., x_s\} \in X \mid (\nexists x_k \in X)\ x_j <_{ts} x_k <_{ts} x_{j+1}, j \in \{1, 2, ..., s-1\}$;
    $Y \leftarrow Y \cup \{X_1\}$;
    $X \leftarrow X - X_1$;
end for
for $y \subset Y$ **do**
    $(y, \leq_{tr}) \leftarrow sortTrust(y, m)$;
end for

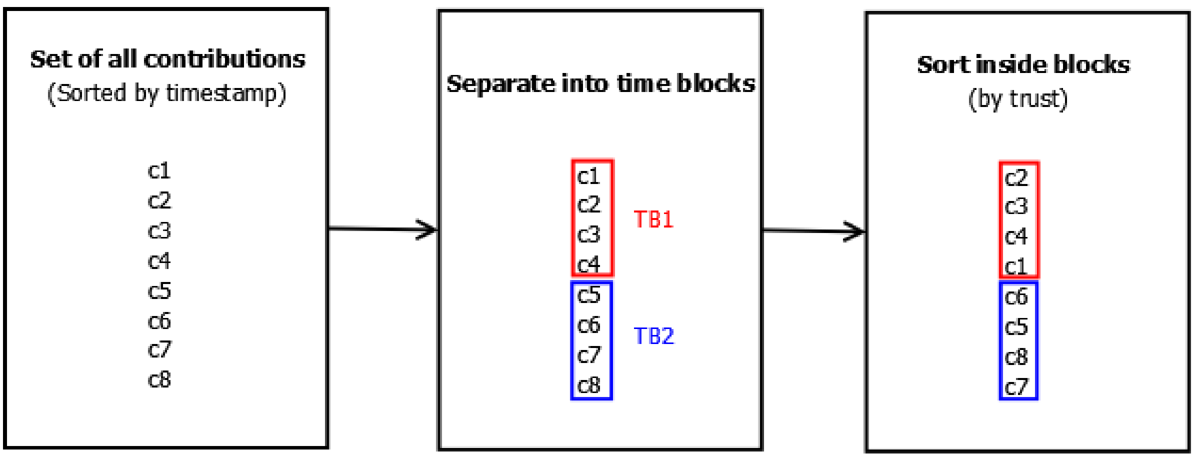

Figure 5.4: Sorting basic functioning

- $<_{ts} \subseteq Timestamp \times Timestamp,\ t_1 <_{ts} t_2 \Leftrightarrow t_1$ comes before than t_2;
- $\leq_{tr} \subseteq (Trust \subseteq \Re) \times (Trust \subseteq \Re),\ t_1 \leq_{tr} t_2 \Leftrightarrow t_1 \leq t_2$;
- $<_{tr} \subseteq (Trust \subseteq \Re) \times (Trust \subseteq \Re),\ t_1 <_{tr} t_2 \Leftrightarrow t_1 < t_2$.

In **Algorithm 3**, we define two *categories* trivially definable from totally ordered sets - $(X, \leq_{ts})$ and $(y, \leq_{tr})$ [26]. First, we have the category $(X, \leq_{ts})$, which is the set of all contributions with an order relation on this contributions timestamps. This set of contributions is **finite**, and the category can be visualized as a **lattice** [6]. In this lattice, we have one minimal element, which can be seen also as the initial object of the respective category. The existence of exactly one minimal element is guaranteed by the Service class, which records one contribution at a time on the database, and therefore each contribution has a timestamp. When we get the initial element of this category, we are actually getting the first contribution. That done, the next $(s - 1)$ contributions are fetched in order (guaranteed by the condition on line 5 in A). Then, a set of several subsets of contributions (each of size s) is formed, and furthermore each of these subsets are sorted by the trust relation previously defined and the mapping m - generating the category - or totally ordered set - $(y, \leq_{tr})$.

In a general explanation, the algorithm first sorts all contributions according to their timestamps. Then, this subset is split in subsets of size s - in the developed system, $s = 4$, because this is the maximum number of contributions that directly fit the screen, in order to make the system usable for the user - and these subsets are sorted internally according to the trust mapping previously explained.

5.2.4 The Contribution Service Class

The Contribution Service Class performs CRUD operations and queries on the tables related to the contribution tables. The most important functions in this class are the ones that perform the queries to fetch:

- all the *father contributions*, which are the ones that appear directly in the system;
- all the *child contributions*, which are the ones that belong to some *father contribution* - that is, the ones that are some comment related to a contribution;
- some single contribution, in order to associate comments and interest to.

All of the *father contributions* are fetched very straightforwardly through the query that finds all contributions. The procedure that explains this is in **Algorithm 4**.

Algorithm 4 $findAllFathers()$

```
contributionList ← findAllFathersQuery();
return contributionsList;
```

The children contributions are fetched as explained in **Algorithm 5**. The function passes as argument to the query a *fatherid* to select the *father contribution*.

The procedure that fetches a single contribution through the corresponding query is explained in **Algorithm 6**. The query in question needs a *contributionid*, which is passed by this procedure.

Algorithm 5 $findAllChildren(fatherId)$

1: $contributionsList \leftarrow findAllChildrenQuery(fahterId);$
2: **return** $contributionsList;$

Algorithm 6 $findSingleContribution(contributionId)$

1: $contribution \leftarrow findContributionQuery(contributionId);$
2: **return** $contribution;$

5.2.5 The Trust Service Class

The Trust Service Class is the one that performs the queries on the database and does the necessary operations over them. For **Algorithm 7**, the procedure for calling the query to retrieve the direct trust from one user to another - explained in **Algorithm 15** - is very straightforward. As there is exactly one path whose size equals one between two nodes in the trust graph, we just have to fetch this trust.

Algorithm 7 $l1t(\alpha, \beta)$

1: $trust \leftarrow levelOneTrustQuery(\alpha, \beta);$
2: **return** $trust;$

As for the indirect trust of level two, we receive from the query a list of pairs whose results are already ordered - explained in **Algorithm 16**. Each pair on the list contains the trust values on the path in question. Then, we just have to fetch the first element (which is a pair) of this list, multiply the pairs elements and multiply the result for the attenuation coefficient - which in this case is $\frac{1}{2}$, as the size of the path is 2.

The generation procedure of the indirect trust of level three is very similar to the one of level two. The query, this time, returns an ordered list of triples. We just have to fetch the first triple, multiply its components and multiply the result for the attenuation coefficient - $\frac{1}{3}$ in this case.

5.2.6 How trust is built up in the system

First of all, we assume that every user of the system trusts no other user in the beginning. Trust is only build up from certain user to another when this first user proves through the system that he is interested in the content shared by the second user - this is expressed when the first user ”likes” or comments the content from the second.

If some user α proves his interest to a user β, for whom α still has no trust, the trust value from α to β is set to 0.5. If α already trusts β to some degree, this trust has to be taken into consideration and be increased.

Then, the formula for trust from agent α to agent β ($t_{\alpha,\beta}$) in function of time (or iteration number) x used in the system, inspired by [39], is:

$$t_{\alpha,\beta}(x) = \begin{cases} 0.5 & \text{if } x = 0 \\ tanh(t_{\alpha,\beta}(x-1) + \sigma \cdot p(\alpha,\beta) + \gamma) & \text{if } x > 0 \end{cases}$$

where:

- $p(\alpha, \beta)$ is the profile similarity between agents α and β;
- σ is the profile similarity importance coefficient;

Algorithm 8 $l2t(\alpha, \beta)$

1: $trustList \leftarrow levelTwoTrustQuery(\alpha, \beta)$;
2: $(t_1, t_2) \leftarrow first(trustList)$;
3: **return** $\frac{1}{2}(t_1 t_2)$

Algorithm 9 $l3t(\alpha, \beta)$

1: $trustList \leftarrow levelThreeTrustQuery(\alpha, \beta)$;
2: $(t_1, t_2, t_3) \leftarrow first(trustList)$;
3: **return** $\frac{1}{3}(t_1 t_2 t_3)$

- γ is the normal increment step.

Within this formula, we assume:

- $\gamma = 0.5$;
- $\sigma = 0.8$.

The γ value has to be chosen in order to determine the speed of the increase of the trust in the course of time. The σ value has to be chosen in order to determine the importance of the profile similarity between the two users, which is still to be explained. With $\sigma = 0.8$ we give a big importance to the profile similarity of patients who use the system, and with $\gamma = 0.5$ we set a normal pace for the increase of trust - this value can be adjusted for different system objectives: if the system requires a certain level of security, for example, trust should be slowly increased, whereas in systems whose requirements do not involve such huge amount of security, trust may increase faster.

5.2.7 Trust transitivity in the system

In the developed system, we consider trust to be transitive to a certain level. More specifically, we consider from direct trust between users until paths of size 3 in the trust graph. The trust from some user to the others is fetched as explained in **Algorithm 10**.

These are some brief explanations of non-trivial assessable names of variables/functions in **Algorithm 10**:

- $l1t(\alpha, \beta)$: level one trust, represents the direct trust from α to β;
- $l2t(\alpha, \beta)$: level two trust, represents the trust in path of size 2 from α to β in the trust directed graph;
- $l3t(\alpha, \beta)$: level three trust, represents the trust in path of size 3 from α to β in the trust directed graph;
- sic: profile similarity importance coefficient;
- $ps(\alpha, \beta)$: profile similarity between α and β.

It is important to point out two mains points of **Algorithm 10**:

Algorithm 10 getTrustValue(α, β)

```
if α = β then
    return 1;
end if
if existsSizeOnePath(α, β) then
    return l1t(α, β) + sic · ps(α, β);
else
    if existsSizeTwoPath(α, β) then
        return l2t(α, β) + (sic · ps(α, β));
    else
        if existsSizeThreePath(α, β) then
            return l3t(α, β) + sic · ps(α, β);
        else
            return 0 + sic · ps(α, β);
        end if
    end if
end if
```

- the algorithm structure privileges trust values based on the proximity of the users, and does not necessarily return the greatest trust path in the graph. That is important to ensure the direct trust from user to user, instead of building ”artificial” trust networks;

- the profile similarity counts both for the definition of trust from one user to another and for the rapid increase of trust from one user to another.

5.2.8 Profile similarity

In the iCare project, each user is associated with certain health profile. These profiles - e.g. diabetes or obesity - gather some of the following measurements - in this work, a subset of the total number of measurements on the system is used:

- Pressure: if the patient has a condition where the pressure should be measured and controlled, this measurement goes into this patient;
  - Systolic;
  - Diastolic.

- Weight: if the patient has to control his weight, this should go into his profile;

- Height: appropriate for when the patient has to control his height;

- Temperature: if temperature measurements are relevant to the user, this should be taken into consideration;

- Glucose: if the patient has glucose issues, this should be monitored;

- Heart rate: should be present in the patients profile if he has some heart-related issue;

- Respiratory rate: for respiratory issues, this measurement should be present in the profile;

- Heart rate in exercise: analogous to heart rate, but during the practice of exercises;
- Number of daily steps: movement is important in almost any health condition, and therefore the number of daily steps must be measured in some profiles.

As an example of concrete profile, we can take *Diabetes* for instance. It gathers systolic and diastolic pressure, glucose, heart rate and number of daily steps. The schema for this formation is in **Figure 5.6**.

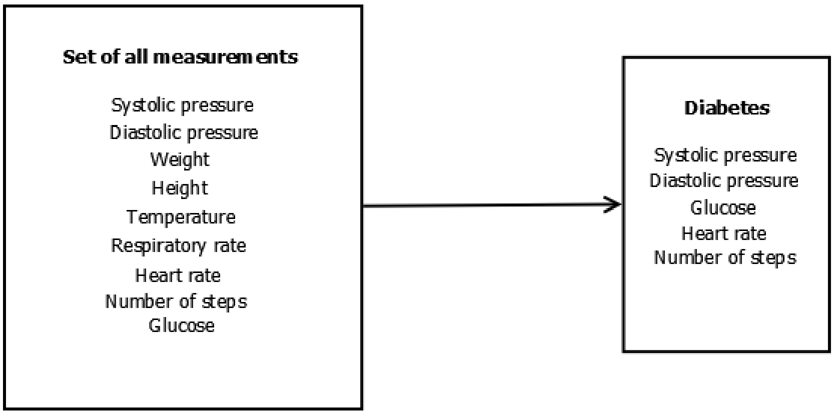

Figure 5.5: Profiles are formed by measurements

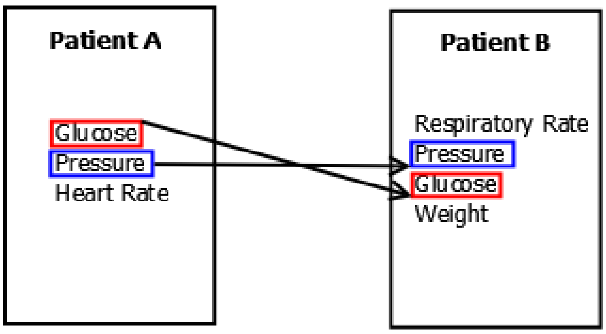

Figure 5.6: Profiles put side by side for comparison

These profiles are planned to be generic health condition profiles, but, in principle, they may be personalized for each patient in the system. That makes room to the comparison between patients profiles. This way, a patient *P* who has a profile with a high degree of similarity to some other patient *Q* may be specially interested in *Q*s contributions.

The comparison between two profiles is done through **Algorithm 11**, where α and β are patients with sets of measurements - i.e. profiles.

To summarize **Algorithm 11**, it takes the number of measure type coincidence from the profiles of both patients and divides it for the total number of distinct profiles. These are the explanations for the variables used:

- p1,p2: first and second profiles, respectively;
- cc: coincidence count, the number of measurement coincidence between the profiles;

Algorithm 11 ps($p1, p2$): Profile similarity between profiles $p1$ and $p2$

if $empty?(p1)$ **or** $empty?(p2)$ **then**
  return 0;
else
  $cc \leftarrow 0$;
  for $m \in p1$ **do**
    if $contains(p2, m)$ **then**
      $cc \leftarrow cc + 1$;
    end if
  end for
  $tdp \leftarrow size(p1) + (size(p2) - cc)$;
  return cc/tdp;
end if

- m: measurement in a profile;
- tdp: total of distinct profiles.

5.2.9 Queries for contributions

There are two main queries for contributions, explained in the following algorithms.

In **Algorithm 12**, we aim to fetch all the contributions that must appear directly in the system. That is, all the contributions that are not comments - all the *father contributions*.

Algorithm 12 Find all fathers

$SELECT\ *$
$FROM\ mbswallcontribution\ AS\ c$
$WHERE\ c.fathercdlcontribution\ IS\ NULL$

In **Algorithm 13**, we aim to fetch all the contributions that have some *father contribution* related to it. That is, we must fetch all the comments on some other contribution. The query needs a *fatherid* to select the related *father contribution*.

Algorithm 13 Find all children

$SELECT\ *$
$FROM\ mbswallcontribution\ AS\ c$
$WHERE\ c.fathercdlcontribution = fatherid$

In **Algorithm 14**, we aim to fetch a single contribution, given by some *contributionid*.

5.2.10 Queries for trust

Considering the table *admusuariotrust* described above, the query to select the direct trust (if it exists) from some user *srcusuario* to some other user *dstusuario* is as explained in **Algorithm 15**.

The query to select the trust of some *srcusuario* to some *dstusuario*, who have a path of size 2 between them in the trust graph is given by **Algorithm 16**. It is important to refer two aspects here:

Algorithm 14 Find single contribution

```
SELECT *
FROM mbswallcontribution AS c
WHERE c.cdlcontribution = contributionid
```

Algorithm 15 Select Level One Trust

```
SELECT t.trust
FROM admusuariotrust AS t
WHERE t.srcusuario = codsrcusuario
AND t.dstusuario = coddstusuario
```

- the fact that there might be more than one path between these users in the trust graph is foreseen by the query. There is an ordering of the results returned, through an *ORDER BY* clause;
- the selection of the best path is done here with the use of the trust from the source user (*srcusuario*) to the intermediary user on the path. That does not necessarily return the path of highest value present on the graph between both users, but the most important factor here is the *trust from the source user on the trust of the second user*.

Algorithm 16 Select Level Two Trust

```
SELECT t1.trust, t2.trust
FROM admusuariotrust AS t1, admusuariotrust AS t2
WHERE t1.dstusuario = t2.srcusuario
AND t1.srcusuario = codsrcusuario
AND t2.dstusuario = coddstusuario
AND t1.srcusuario <> t1.dstusuario
AND t2.srcusuario <> t2.dstusuario
ORDER BY t1.trust DESC
```

The query to select the trust of some *srcusuario* to some *dstusuario*, who have a path of size 3 between them in the trust graph is given by **Algorithm 17**. It is important to point out here that both the aspects brought up in the explanation of **Algorithm 17** are analogous in this item. The ”best” path is also chosen based on the trust from the source user on the trust of the first intermediary user.

Algorithm 17 Select Level Three Trust

```
SELECT t1.trust, t2.trust, t3.trust
FROM admusuariotrust AS t1,
     admusuariotrust AS t2,
     admusuariotrust AS t3
WHERE t1.dstusuario = t2.srcusuario
AND t2.dstusuario = t3.srcusuario
AND t1.srcusuario = codsrcusuario
AND t3.dstusuario = coddstusuario
AND t1.srcusuario <> t1.dstusuario
AND t2.srcusuario <> t2.dstusuario
AND t3.srcusuario <> t3.dstusuario
AND t1.srcusuario <> t2.dstusuario
ORDER BY t1.trust DESC
```

6 VALIDATION

In order to easily achieve and prove the earlier mentioned goals, it is important to set some ground rules for testing and validation. The methodology for these two matters are approached in this section.

6.1 Testing Methodology

As the results of this work are tightly related to the response from the user, we need to find a way to validate them in relation to the users experience with the system. In order to achieve that, a series of satisfaction forms must be elaborated to fetch quantitative response from them. These forms are explained briefly in the following subsection. Also, measures from each individual users usage of the system will be taken into consideration. A good sample of usage allied with a satisfaction form that fetches quantitative results should expose and prove the final result of the work.

6.1.1 Testing Forms

The validation of the present work is all about approximating the systems response to the ideal configuration for the user. In this sense, the validation will proceed in this fashion:

- A certain fixed number of users will be gathered to use the system constantly for a given period of time: in order to create a meaningful measurement, twelve randomly chosen people will use the system for 10 days, which is a considerable time for building trust relations, posting at least one message per day and being free to relate them to the others by *liking* or *commenting* other posts;
- The systems status after this period of testing will be retrieved for each user;
- Each user will, then, give a value of importance to each of the messages on the system, i.e., will create its own ranking for the messages;
- This ranking will be taken into consideration as optimal, and the mean squared error (MSE) will be taken from this optimal to a *timestamp-organized* and to a *trust-organized* list of messages, for reasons of comparison;
- If the MSE for the trust-organized list of messages in relation to the users optimal is lower than the same relation for the the timestamp-organized list, we have a concrete proof that the trust-based system has a serious contribution in the concern of fast and direct data acquisition applied to a usable telemedicine software.

Table 6.1: Validation method: contribution set S_j

Contributor	Contribution	I_i	TS_i	TR_i
$Contributor_1$	C_1	I_1	TS_1	TR_1
$Contributor_2$	C_2	I_2	TS_2	TR_2
$Contributor_3$	C_3	I_3	TS_3	TR_3
$Contributor_4$	C_4	I_4	TS_4	TR_4

The invitation letter for contributors can be seen in **Appendix A**. It tells briefly the invited person about the overall of the system and how the testing will proceed - as explained in the items right above.

After the twelve chosen testers have spent ten (10) days testing the system - i.e., posting contributions on the system and relating themselves to the others by "liking" and/or commenting other contributions - the validation part comes. In order to prove the effectiveness of the trust system, the users must fill in a form of validation.

The validation form can be seen in **Appendix B**. It consists of nine (9) sets of four (4) contributions, as seen on each page of the system. In the form, the user must evaluate each contribution in a set with a number that represents the importance of the contribution in question to him. To do that, this user gives a value from 1 to 4 to each contribution in each set of contributions. This value is such that 1 means the contribution is very important and 4 means the contribution is very unimportant.

After this data has been provided by the user, we compare the "ideal" sorting of the information to the sorting by time and trust and the sorting by time alone. The comparison is done, as already explained, through the MSE from the trust based system and the timestamp based system with the ideal sorting.

As an example, let us take a generic contribution C. C is classified, within a group of four contributions G, as the most important - that is, it has value 1. Let us now suppose that, in the timestamp based system, this same contribution comes in second place - that is, with value 2 - while in the trust based system it comes in first place - i.e., as in the ideal sorting, it has value 1. Then, the error from the trust based system to the ideal system is 0, while the error from the timestamp based system is 1. The errors of each of the following contributions for each set are then raised individually to the power of 2, summed up and divided by the total of contributions. The system for which this summed error is smaller is then considered the best.

As it can be seen in **Table 6.1**, we have a set of 4 contributions. On the validation phase, we have 9 sets of 4 contributions each. The validation will then work in this fashion:

- Let i_i be the ideal sorting for contribution i;
- Let ts_i and tr_i be the sorting according to timestamp and trust, respectively;
- We take the MSE from each set S_j of contributions and for each sorting, i.e.

$$MSE_{ts}(S_j) = \frac{1}{4} \cdot \sum_{i=1}^{4} (I_i - TS_i)^2$$

$$MSE_{tr}(S_j) = \frac{1}{4} \cdot \sum_{i=1}^{4} (I_i - TR_i)^2$$

- We then take that MSE for both sorting types and take the sum of them concerning all the contribution groups, i.e.

$$Sum_{ts} = \sum_{j=1}^{9} MSE_{ts}(S_j)$$

$$Sum_{tr} = \sum_{j=1}^{9} MSE_{tr}(S_j)$$

That way, we have a sum of the mean squared errors for the contribution sets. We can compare this sums for both sorting methods and compare which of them approximates itself more from the ideal sorting.

The mean squared error is taken in order to value the closeness of the sorting to the ideal sorting. In this sense, if a contribution c is classified as very important ($i_c = 1$) and a given sorting s_c gives a very unimportant classification to that contribution (e.g. $s_c = 4$), this will be more severely penalized.

7 RESULTS

In this chapter we provide the results of the research performed. First, some general results are presented concerning the final state of the testing. Then, the specific validation is presented, showing the results of the trust method in comparison with the "ideal" method and the timestamp method.

7.1 Overall results

In **Figure 7.1** and **Figure 7.2**, the edges with start and end on the same node - which represent the trust everyone has in himself - as well as the edge weights - which represent the trust - are omitted for reasons of simplicity. The arcs of the resulting trust graph are shown explicitly in **Table 7.1**, with trustor, trustee and the trust in question. The graph has 12 nodes - the users - and 65 edges - representing trust relations.

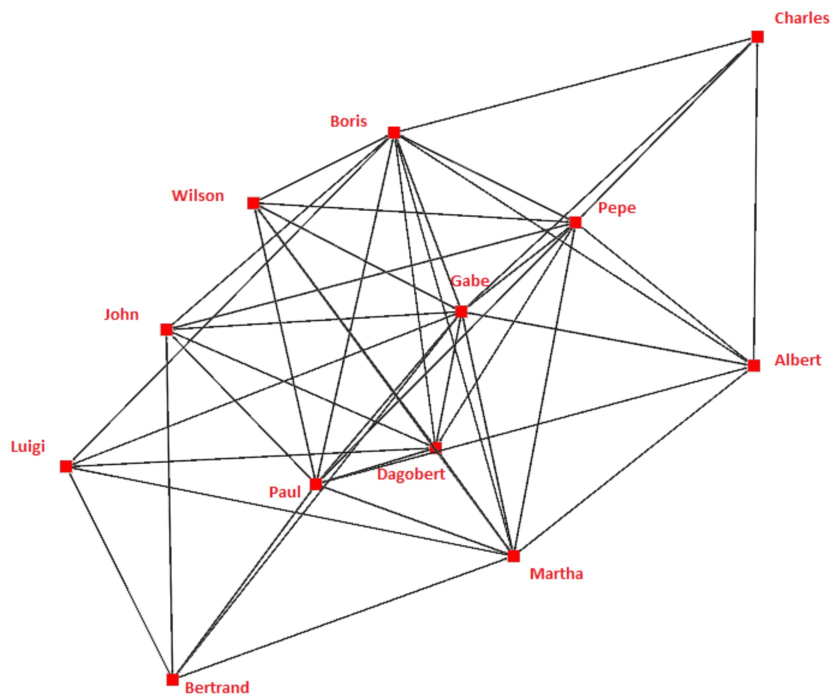

Figure 7.1: Trust graph at the end of the testing period

It is very interesting to observe a very common characteristic of social networks on the resulting graph - the forming of *communities*. In the context of this work, we have a

Table 7.1: The trust graph

Trustor	Trustee	Trust
Gabe	Paul	0.915007629984323
Boris	Gabe	0.956624367395255
Boris	Paul	0.89572394153112
Paul	Gabe	0.911739768692906
Gabe	Boris	0.956624367395255
Wilson	Gabe	0.500000000000000
Wilson	Boris	0.890618000015555
Martha	Boris	0.770703296404525
Martha	Gabe	0.891869175528906
Martha	Paul	0.890618000015555
Gabe	Martha	0.891927051404596
Paul	Martha	0.863289215128892
Dagobert	Boris	0.950870136396236
Dagobert	Martha	0.890618000015555
Pepe	Gabe	0.500000000000000
Bertrand	Martha	0.863289215128892
Pepe	Martha	0.500000000000000
John	Bertrand	0.740525631113721
Pepe	Boris	0.500000000000000
Pepe	Paul	0.914992940924727
Gabe	Pepe	0.956620711355873
Gabe	John	0.500000000000000
Gabe	Bertrand	0.810296507289733
Boris	Pepe	0.500000000000000
Charles	Pepe	0.500000000000000
Albert	Gabe	0.500000000000000
Albert	Charles	0.500000000000000
Pepe	Charles	0.500000000000000
Gabe	Charles	0.500000000000000
Gabe	Albert	0.893612025692537
Dagobert	Gabe	0.500000000000000
Paul	Albert	0.810296507289733
Paul	Bertrand	0.500000000000000
Paul	Pepe	0.500000000000000
Paul	Boris	0.500000000000000
Martha	Albert	0.500000000000000
Martha	Pepe	0.891651498655821
Pepe	John	0.815951104486394
John	Pepe	0.500000000000000
Pepe	Dagobert	0.500000000000000
Gabe	Dagobert	0.500000000000000
Paul	John	0.700828415184291
Wilson	Dagobert	0.500000000000000
Wilson	Martha	0.500000000000000
Wilson	Paul	0.854945134501407
Wilson	Pepe	0.500000000000000
Luigi	Dagobert	0.500000000000000
Gabe	Luigi	0.500000000000000
John	Dagobert	0.500000000000000
Boris	Luigi	0.500000000000000
Boris	Albert	0.500000000000000
Boris	Charles	0.500000000000000
Pepe	Albert	0.500000000000000
John	Boris	0.500000000000000
Martha	Luigi	0.500000000000000
Martha	Dagobert	0.500000000000000
Martha	Wilson	0.500000000000000
Paul	Dagobert	0.810296507289733
Paul	Wilson	0.500000000000000
Bertrand	Luigi	0.500000000000000
Bertrand	John	0.624314015922915
Bertrand	Paul	0.893612025692537
Gabe	Wilson	0.500000000000000
Dagobert	Paul	0.500000000000000
Dagobert	Wilson	0.500000000000000

smaller graph, with comparably few nodes and edges in relation to big application graphs - e.g. cities or real large social networks. However, we can see that some nodes of the graph are more intricately connected. In this context, the users **Paul**, **Dagobert**, **Pepe**, **Wilson**, **John** and **Luigi** and strongly connected to each other, forming the community highlighted in green. Also the users **Martha**, **Gabe** and **Bertrand** are closely related to each other, forming the community highlighted in red. The remaining users - **Boris**, **Charles** and **Albert** - the community highlighted in blue. That is a very common phenomenon in social networks [19] [35], which differ them much from random graphs [8]. The algorithm used to find the communities was the one described in [7]. The modularity found by the algorithm was 0.055, which is rather small. This can be explained by the reduced characteristics of the network in comparison to large social networks.

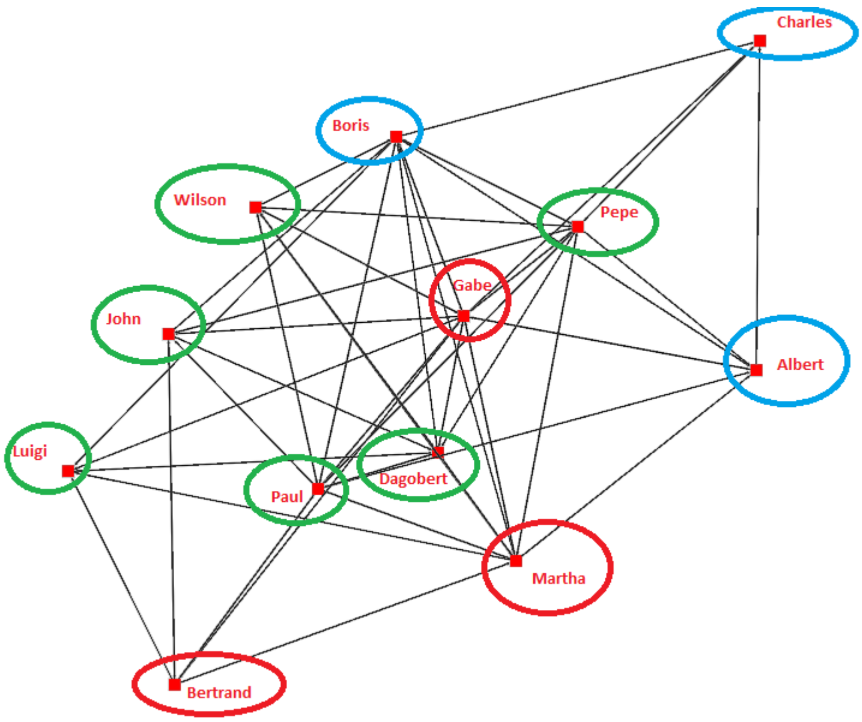

Figure 7.2: Communities on the trust graph

The average path length found for the resulting trust graph is approximately 1.58 - through the algorithm in [10]. The clustering coefficient of the graph in question is 0.621 - according to the algorithm in [24].

In **Table 7.2**, the clustering coefficient (CC) and the average path size (APS) for random graphs with 12 nodes - i.e. the same number of nodes than the resulting trust graph - are shown. The algorithms used were the same used in the trust graph case. In the table, p is the probability to which two nodes of the graph would be connected. As we can see, the trust graph achieved a higher clustering coefficient than the random graphs generated - even those with high probability. That means that the connections of the trust graph are very locally dense, while the random graph - which does not constitute a small-world situation - is globally dense. In addition, the average path size is not much different from the trust graph to the random graphs generated. That means that, even the trust graph being locally dense, nodes that are located outside a community can be achieved through a relatively small number of hops. These data can help us conclude that the resulting trust

Table 7.2: Random graphs with 12 nodes

$p = 0.3$		$p = 0.5$		$p = 0.7$		$p = 0.9$	
CC	**APS**	**CC**	**APS**	**CC**	**APS**	**CC**	**APS**
0.151	1.455	0.229	1.33	0.339	1.267	0.459	1.076

graph of the testing and validation phase constitutes a small-world situation, which is a strong characteristic, as already said, of social networks.

7.1.1 Reputation and the PageRank algorithm

One more aspect to refer is the reputation - concept which was not used in the present work, but still worth pointing out - created by appreciated users. In the trust graph, we can take the node **Pepe** as an example. The user **Pepe** was very active during the testing phase, with many - and relevant - posts. As a result of that strong and relevant participation, many of the other users ended up creating a trust relation with him. Such reputation can be measured through the *PageRank* algorithm. It is important to refer here that this algorithm has better effect on larger graphs, and the reputation relation is specially directed to these cases.

The PageRank algorithm has been brought to the surface in order to calculate the relevance of a certain node of the Web graph [30]. The Web graph is the conceptualization of the World Wide Web as a graph, where webpages are nodes and links between webpages are edges [12]. The concept here is that the most valuable page is the one which is visited the most when *crawling* on a graph. The crawling is done with respect to the weight of the edges, where the probability of going from a node to the other is proportional to the weight of the edge that link them both.

The **Table 7.3** shows the result of the PageRank algorithm - from [11] - for different values of α. There, α is the probability that the walk on the graph through the PageRank algorithm will restart in a random node. The basic PageRank functioning [30] is presented in **Algorithm 18**.

Algorithm 18 Basic PageRank functioning

let $G(V, E)$ be a web-graph
visit node $i \in V$
if dead end is reached **then**
    jump to a random page
else
    with probability α jump to a random page
    with probability $(1 - \alpha)$ follow a link based on weights
end if

For all the provided values of α, the user **Pepe** has the biggest rank. This means that he has the biggest reputation among other users in the trust network.

The PageRank algorithm has a good application in finding reputation in a social network because it sorts the nodes of the social network according to their relevance. On a trust network, we can think that relevance of a certain node is the amount of trust that the other nodes put on it. That being, it may serve well as the indicator of reputation for that node inside the social network.

Table 7.3: PageRank algorithm and reputation

$\alpha = 0.2$		$\alpha = 0.4$		$\alpha = 0.6$		$\alpha = 0.8$	
Pepe	0.097	Pepe	0.109	Pepe	0.12	Pepe	0.13
Dagobert	0.094	Dagobert	0.101	Gabe	0.108	Gabe	0.118
Gabe	0.091	Gabe	0.099	Dagobert	0.105	Paul	0.112
Paul	0.09	Paul	0.097	Paul	0.105	Dagobert	0.105
Boris	0.087	Boris	0.0915	Boris	0.096	Boris	0.101
Martha	0.085	Martha	0.087	Martha	0.089	Martha	0.09
Charles	0.080	Charles	0.076	Charles	0.073	Charles	0.071
Albert	0.077	Albert	0.072	Albert	0.069	Albert	0.067
John	0.076	John	0.07	John	0.065	John	0.059
Bertrand	0.075	Bertrand	0.066	Bertrand	0.058	Bertrand	0.049
Luigi	0.074	Luigi	0.065	Wilson	0.057	Wilson	0.049
Wilson	0.074	Wilson	0.065	Luigi	0.056	Luigi	0.047

Table 7.4: Validation: mean-squared error

User	Sum_{ts}	Sum_{tr}
Paul	27.0	**13.0**
Dagobert	27.0	**17.0**
Gabe	**22.5**	26.0
Pepe	**21.5**	23.0
Charles	**18.0**	20.0
Albert	23.5	**20.0**
Boris	**16.5**	24.5
Bertrand	30.0	**18.0**
Wilson	28.3	**20.5**
John	25.5	**18.0**
Martha	**22.0**	22.5
Mean value	≈ 23.8	≈ 20.2

7.2 Validation results

When comparing both the timestamp-based and the trust-based sorting methods, we can see that the trust-based has a lower sum of mean squared errors in general, which makes it a better fit for the ideal sorting. It is important to elucidate that only 11 out of the 12 testers agreed to participate on the validation (all 12 agreed on testing).

It is important to point out two facts from the results:

- The trust-based algorithm overcame the timestamp algorithm in most of the cases - 6 out of 11;
- In the cases in which it did not overcome, the error difference comparing both methods is in general not too large;
- In the cases in which it does overcome, the differences are more considerable;
- An overall conclusion on the better fitting to the ideal sorting from the trust-based sorting in comparison to the timestamp sorting can be drawn from the mean of the errors, present in **Table 7.4**, which are $Mean_{ts} \approx 23.8$ for the timestamp sorting and $Mean_{tr} \approx 20.2$ for the trust sorting.

Furthermore, we can conclude that the trust-based method proposed in this work tends to shorten distances from the actual sorting to the ideal sorting.

If we observe **Table 7.5**, we can see that contributions which are too many positions apart from the ideal sorting position are severely penalized. Let us take the contribution C_1 for matters of elucidation. C_1 has an ideal rank of 1, i.e. it is the most important. Suppose now, as shown on the table, that the timestamp-based sorting gives a rank of 4 to that contribution. The simple error would return a value of $(4 - 1) = 3$, whilst the squared error returns a value of $(4 - 1)^2 = 9$. The penalty for the bad rank given by some algorithm is then much higher for the mean squared error than for the simple mean error.

Also worth observing is the fact that the higher the distance is to the ideal, higher is also the error. This rule does not follow a linear dependence, but a quadratic one. Then, the higher distances are even more critical than they would be for the simple mean error.

Table 7.5: Validation method: error example for some set S_j

Contribution	I_i	TS_i	TR_i
C_1	1	4	2
C_2	2	3	1
C_3	3	2	4
C_4	4	1	3
	MSE	5	1

In **Table 7.5**, we can see that the ranking given by the trust-based algorithm provides a lower error by shortening the distances from the ideal ranking to the given algorithm ranking.

There are some individual results worth paying attention to. The only result in which the timestamp sorting algorithm beats the trust sorting algorithm to a great level - i.e. the ratio of Sum_{tr}/Sum_{ts} is large - is for user **Boris**. As for the other side of this statement, the times when the trust based algorithm ends up outstanding the timestamp-based sorting algorithm are multiple, and the ratios are quite large.

Let us take into account the users **Paul** and **Dagobert**. We can observe in **Table 7.4** that the errors are the following:

- *Paul*:
  - Timestamp-based algorithm: $Sum_{ts} = 27$;
  - Trust-based algorithm: $Sum_{tr} = 13$;
- *Dagobert*:
  - Timestamp-based algorithm: $Sum_{ts} = 27$;
  - Trust-based algorithm: $Sum_{tr} = 17$;

For **Paul**, we can see that the trust-based method provided a fitting to the ideal ranking that is more than *twice better* than the fitting for the timestamp-based method:

$$\frac{Sum_{tr}}{Sum_{ts}} \approx 0.481$$

For **Dagobert**, the fitting of the trust-based method is also much tighter than the fitting of the timestamp-based algorithm. It is not as comparably tight as in the case of **Paul**, but is almost two times better than the timestamp method:

$$\frac{Sum_{tr}}{Sum_{ts}} \approx 0.63$$

There are cases where the fitting proposition for both methods are quite similar. Concerning this, let us take the case of user **Martha** as an example:

- Timestamp-based method: $Sum_{ts} = 22$;
- Trust-based method: $Sum_{tr} = 22.5$;

In this case, the error on the errors is almost negligible, and both methods are considered almost equally efficient:

$$\frac{Sum_{tr} - Sum_{ts}}{Sum_{tr}} \approx 0.022$$

Based on the above results, we can assume that the trust-based method implemented and validated in this work has been better than a timestamp-based method for the given testing and validation environment.

8 CONCLUSION AND FUTURE WORK

As a first important conclusion, it is interesting to point out that the resulting trust network from the testing phase reflected the structure of a real social network, as already expected. Technically speaking, it showed the following social network characteristics:

- Forming of communities: three different communities were formed on the resulting graph, confirming the *locally dense* and *globally sparse* property of such networks;
- High clustering coefficient: in comparison to random graphs generated to the same vertex set, the resulting trust graph presented a high clustering coefficient, what strengthens the concept of locally dense communities;
- Low average path size: still in comparison with these random graphs, the average path size for the trust graph is relatively low, what makes it possible to go from one community to another with a small number of hops;

Still a point worth considering is the application of the PageRank algorithm on the graph to discover the user with the biggest reputation. This reputation comes then into action on the trusting decisions of an agent towards some contribution present on the system.

In terms of the proposed validation and considering our sample from 12 people actively contributing on the social network over 10 days, we say that the proposed trust-based sorting algorithm has been shown itself a better fit for the ideal sorting algorithm in comparison to one broadly used sorting algorithm, which is the timestamp-based one. Both in overall and in individual results, the algorithm presented in this work has shown a smaller error in relation to the ranking generated by the ”ideal” sorting algorithm.

As already referred in the **Results** chapter, some important observations on the validation results are the following:

- For the majority of the users (6 out of 11), the trust-based method proved to provide a better fit to the ideal sorting in comparison to the timestamp-based method;
- For the cases in which the timestamp-based method presented a better fit to the ideal sorting, the difference comparing the errors of the trust-based and the timestamp-based models is relatively small. This means that, for these cases, both fits provided are almost equal;
- For the cases in which the trust-based method provided a better fit in comparison with the timestamp-based method, the differences in the errors of both methods

are quite large. That means, aggregating the results of the last item, that the trust-based method tends to provide a better or ”almost equal” fit to the ideal sorting in comparison with the timestamp-based model;

- The overall performance of the trust-based system can be seen in the mean value of the summing of the errors of every patient. This is the mean error for the each model. The mean error for the trust model is smaller than the mean error for the timestamp model.

To summarize, the implemented and validated trust-based social network has shown to be of relevance. It provides a good framework to help the treatment of homecare patients and an information sorting method that provides a good fit to the ideal sorting for each user.

8.1 Future work

In the future, the application of the system on a large scale is a possible work path. It may be very interesting to observe the benefits of social computing and Information and Communication Technology (ICT) in a broader spectrum in health care. If the system were applied to a broader range - that is, to many users - we could possibly observe more easily some common happenings in large social networks. When a social network is large enough, it is most commonly composed of several dense communities connected to each other by few nodes, making it possible to navigate through communities with a relatively small number of hops - *small world* - and making it very similar to a Watts-Strogatz graph [41].

Another suggestion worth considering is applying the concept of reputation during the attribution and building of trust. If certain user has a good reputation, than it would be more likely that another random user would trust him. That can be measured by the *Page Rank* algorithm, and can improve the similarity of the trust network resulting of this work to a large scale social network.

The expansion of the maximum level of transitivity of trust is another point to build upon. In the present work, we deal with transitivity to a level of three - that is, the the size of the path on the trust graph is at most three. Within this context, a path of level three was sufficient, but on real social networks, we may have a larger world, and the expansion of trust transitivity can give us many more insights on the behaviour of agents with concern to trust as well as a possible closer mapping from real trust to the mathematical formalization of trust.

In the current trust-based model, two messages from the same user are sorted deterministically according to their timestamps. The most recent messages appear first. That makes the algorithm very similar to the timestamp-based in some cases when a certain user makes too many consecutive posts, for example. That logically approximates the error of the trust-based model to the error of the timestamp-based model. In the future, it is important to optimize this problem. One suggestion is to attach subjects or contexts to messages or even mining relevant data from them. These data can then be compared to the current user’s profile. That way, the ranking from the trust-based method for consecutive messages from the same user would possibly differ from the timestamp-based model. That can lead us to two possibilities:

- The error can rise: if the relevant content from the message in question are erroneously fetched, the error of the trust-based model can rise much above the error of the timestamp-based method;

- The error can sink: if the relevant content of the message is fetched correctly and is properly compared to the current user's profile, we will probably have a better sorting of messages of the trust-based method compared to the timestamp-based method.

To summarize, trust in social networks is a field worth researching on. Many aspects can still be optimized, from the attribution of trust passing through reputation and information sorting algorithms.

APPENDIX A INVITATION LETTER FOR TESTERS

Dear friend,

Through this letter I invite you to take part in the testing of the contribution tool of the TV-Care sytem, which was developed for my graduation work in the course of Computer Science. The TV-Care is a remote homecare system, that is, a system used to monitor the health status of patients who are treated in their homes, remotely observed by a specialist.

The contribution tool consists of a virtual wall where users of the system (mainly patients) post informations they consider to be useful and/or interesting about general subjects, including the main focus of the system, which is health care. This tool has been developed in a way that the most important messages should appear easily to the user, in a kind of a personalized sorting.

In order for this tool to be validated, it is interesting that some people make use of it during a given period of time, so that afterwards we can take some conclusions about its utility. In this context, I invite you to take part, during the period of 10 (ten) days, on the validation of the system. Your task in these days would be to contribute with at least one message per day on the wall. This task will not take you more than five minutes, and the messages may have any character, except the ones that may be offensive in some way. Each and every participation from the user concerning commenting of clicking ”I like it” on some contribution is highly encouraged, once the system depends on the interactions among the users. Therefore, if you find some contribution interesting, it is strongly recommended that you express you interest in such information via these virtual tools - ”I like it” and comments.

In case you are interested in taking part on this experiment, please notify me. That done, additional informations will be sent to you so that you registering can be done and the testing phase may be initiated.

I am grateful for your attention and hope there is interest from you,
Matheus Priebe Bertram

APPENDIX B VALIDATION FORM

Dear tester,

Please fill in the tables below in the following manner:

- In each table, sort the messages that are more relevant to **YOU**, not depending on the type of message;
- Do not feel shy to sort some message of yours as the most important of the block in case you really think so;
- Give a value from 1 to 4 to each line of the table, where **1 = very important** and **4 = very unimportant**. The values inside every table must not repeat themselves.

Your name:

Name	Contribution	Value
Paul	Este site mostra alguns benefícios e dá dicas sobre corrida, tanto para iniciantes quanto para aqueles já praticantes: http://sportlife.terra.com.br/index.asp?codc=1028	
Gabe	Veja abaixo os boatos e mitos mais frequentes sobre a Coca-Cola http://www.cocacolabrasil.com.br/boatos_mitos.asp?categoria=1	
Gabe	Portal sobre Sáude do Governo Federal: http://portalsaude.saude.gov.br/portalsaude/index.cfm	
Gabe	Hábitos Saudáveis: http://belezaesaude.dae.com.br/habitos-saudaveis/	

Figure B.1: Validation table 1

Name	Contribution	Value
Boris	http://extra.globo.com/noticias/mundo/com-368-kg-britanico-o-homem-mais-gordo-do-mundo-3958443.html	
Boris	http://extra.globo.com/noticias/mundo/com-368-kg-britanico-o-homem-mais-gordo-do-mundo-3958443.html	
Boris	http://www.clubedoscarecas.com.br/notas_pagina.php?det=38	
Martha	O site http://www.minhavida.com.br/ é bem interessante, além de artigos e vídeos, tem testes interativos interessantes sobre expectativa de vida, peso ideal, se você está pronto(a) para começar a correr. Vale a pena conferir mais dicas saudáveis também ;-	

Figure B.2: Validation table 2

Name	Contribution	Value
Martha	Meu teste de exectativa de vida deu 84 anos, superior a média da mulher brasileira que é de 78 anos. Bem, nem cheguei na metade ainda, então, "vamo que vamooooo". Façam o teste e avisem como foram :-D	
Martha	Leiam esta matéria e diminuam o consumo de refrigerantes :-) http://www.dicasnutricionais.com.br/coca-cola-zero-e-proibida-nos-eua-e-no-brasil-sete-refrigerantes-tem-substancia-cancerigena/	
Martha	Excesso de peso pode comprometer o funcionamento do cérebro... http://www.dicasnutricionais.com.br/o-excesso-de-peso-pode-comprometer-o-cerebro/	
Gabe	Ponderando os benefícios da gordura - http://www.guiame.com.br/ntc/ponderando-os-beneficios-da-gordura.html	

Figure B.3: Validation table 3

Name	Contribution	Value
Gabe	O ganho de peso não é só um problema de quem come demais, mas também de quem gasta energia de menos. E nessa conta matemática, quem diria, até o ar-condicionado ajuda no ganho de peso.	
Bertrand	McDonald's troca receita de hamburger após denúncias. É bom saber bem ao certo o que tem nos alimentos que comemos! http://economia.terra.com.br/noticias/noticia.aspx?idNoticia=201201261711_TRR_80776024	
John	Falha genética aumenta o risco de grave doença do coração: http://oglobo.globo.com/saude/falha-genetica-aumenta-risco-de-grave-doenca-do-coracao-3741531	
Pepe	Pressão alta e corrida - Hipertenso pode correr? Exercício de alta intensidade não são aconselháveis para portadores de pressão alta. http://www.copacabanarunners.net/perg0901.html	

Figure B.4: Validation table 4

Name	Contribution	Value
Gabe	Benefício da cerveja e do vinho: http://veja.abril.com.br/noticia/saude/especialista-tira-duvidas-sobre-os-beneficios-da-cerveja-e-do-vinho	
Pepe	Os Benefícios do Suco de Uva - http://www.vinicolagaribaldi.com.br/pt/saude/os-beneficios-do-suco-de-uva/	
Charles	Aplicativo para celular produz um ECG a partir do seu dedo https://market.android.com/details?id=com.azumio.instantheartrate.full	
Albert	Os Benefícios da Cerveja (ela também ajuda a prevenir o câncer) http://bad-skipit.blogspot.com/2005/11/benefcios-da-cerveja.html	

Figure B.5: Validation table 5

Name	Contribution	Value
Paul	Descubra sua idade interior: http://www.idadeinterior.com.br/. A minha é 19 anos :)	
Bertrand	Por quê fracionar as refeições Dividir a alimentação em 3 refeições principais e 3 lanches intermediários. Isso evita que o indivíduo fique beliscando entre as refeições, já que irá consumir pequenos lanches, aumenta o trabalho intestinal, pois have	
Martha	No link http://www.cdof.com.br/testes11.htm, você pode fazer o Questionário da Saúde, que dá sua idade de saúde (a minha 20 anos... hehehe) e de expectativa (95 anos, ganhei mais uma década nesse teste)	
John	Dieta muito calórica causaria perda de memória? http://g1.globo.com/ciencia-e-saude/noticia/2012/02/idosos-com-dieta-calorica-tem-risco-alto-de-perda-de-memoria-diz-estudo.html	

Figure B.6: Validation table 6

Name	Contribution	Value
Martha	Qual a idade do teu cérebro? Faça o teste em http://sorisomail.com/partilha/118799.html . Interessante, bah... o negocinho é rápido, mas interessante para exercitar as "conexões". Estou com 37 anos de cérebro :-(Acho que preciso exercitar a massa cinzent	
Martha	a mais um pouco, estou 3 anos atrás do que deveria... hehehe.	
Pepe	Pra brincar um pouco: Gaúchos que cuidam da sua saúde vivem mais que brazileiros - http://www.obairrista.com/noticia&codigo=633	
Gabe	http://zerohora.clicrbs.com.br/rs/geral/noticia/2012/02/menino-morto-por-cachorro-e-velado-na-capela-municipal-de-capao-da-canoa-3664460.html	

Figure B.7: Validation table 7

Name	Contribution	Value
Dagobert	Dicas para melhorar a memória: http://www.bircham.org/guia-de-estudos/conselhos/melhorar-memoria.html . Interessante, diz que "80% do que estudamos, perdemos depois de 24 horas".	
Luigi	Depois dos 40 é mais saudável ser gordinho. http://hypescience.com/22689-aos-40-e-mais-saudavel-ser-gordinho/	
Albert	Cuidem-se no carnaval, e bebam bastante água. http://www.brasilescola.com/carnaval/cuidados-com-o-corpo-durante-o-carnaval.htm	
Boris	http://veja.abril.com.br/090507/p_128.shtml	

Figure B.8: Validation table 8

Name	Contribution	Value
John	Corramos! http://g1.globo.com/ciencia-e-saude/noticia/2012/02/andar-lento-pode-prever-alzheimer-em-pessoas-acima-de-60-anos-diz-estudo.html	
Wilson	Especial do Globo Repórter sobre qualidade de vida. http://www.youtube.com/watch?v=I5TmkW6ur80	
Wilson	Olha o oposto do gordo... Que tal um relacionamento do borba com essa guria?	
Paul	Boas dicas sobre aquecimento para pessoas que gostam de correr: http://www.copacabanarunners.net/walk7.html	

Figure B.9: Validation table 9

REFERENCES

[1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, page 9 pp. vol.1, jan. 2000.

[2] Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. Four degrees of separation. *CoRR*, abs/1111.4570, 2011.

[3] Maria Lucia Kroef Barbosa, Valter Roesler, Alexandro Bordignon, Johannes Doll, and Eliseo Berni Reategui. I-care - a health promotion system for active ageing. 2011.

[4] Joyce Berg, John Dickhaut, and Kevin McCabe. Trust, reciprocity, and social history. *Games and Economic Behavior*, 10(1):122 – 142, 1995.

[5] L. F. Berkman and T. Glass. Social integration, social networks, social support, and health. 2000.

[6] Garrett Birkhoff. *Lattice theory*, volume 25. 1995.

[7] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008.

[8] Béla Bollobás. *Random Graphs*. Cambridge University Press, second edition, 2001.

[9] J.A. Bondy and U.S.R. Murty. *Graph Theoy with Applications*. 1976.

[10] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[11] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Proceedings of the seventh International Conference on the World Wide Web*, pages 107–117, 1998.

[12] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Computer Networks*, 33(1–6):309 – 320, 2000.

[13] Vanevar Bush. As we may think. *Atlantic Monthly*, 176(1):101–108, 1945.

[14] Norman L. Chervany. The meanings of trust. *MISRC Working Papers Series*, 1996.

[15] Nicholas A. Christakis and James H. Fowler. The spread of obesity in a large social network over 32 years. *New England Journal of Medicine*, 357(4):370–379, 2007.

[16] Guilherme Lazzarotto de Lima. Classificacao automatica de batidas de eletrocardiogramas. *Graduation work - Universidade Federal do Rio Grande do Sul*, 2011.

[17] Horacio Arruda Falcao. Home care - uma alternativa ao atendimento da saude. *Medicina On Line - Revista Virtual de Medicina*, 2:1, 2010.

[18] Diego Gambetta. *Trust: Making and Breaking Cooperative Relations*. Blackwell Pub, 1990.

[19] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[20] Lynne Grewe. *OpenSocial Network Programming*. 2009.

[21] O. Hanseth and M. Aanestad. Design as bootstrapping. on the evolution of ict networks in health care. *Methods of information in medicine*, page 1, 2003.

[22] Anelise Jantsch, Eliseo Reategui, Maria Lucia Kroeff Barbosa, and Valter Roesler. Proposta de uma interface simplificada para o tvcare: homecare via tv digital para a terceira idade. *VI Congresso Ibero-americano de Telemática*, 2011.

[23] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 137–146, New York, NY, USA, 2003. ACM.

[24] Matthieu Latapy. Main-memory triangle computations for very large (sparse (power-law)) graphs. *Theoretical Computer Science (TCS)*, 407(1-3):458–473, 2008.

[25] Stephen Paul Marsh. *Formalising Trust as a Computational Concept*. Ph.D. Thesis, University of Stirling, 1994.

[26] Paulo Blauth Menezes and Edward Hermann Haeuler. *Teoria das Categorias para Ciência da Computação*. Second edition, 2006.

[27] Steve Milgram. The small-world problem. *Psychology Today*, 1(1):61–67, 1967.

[28] Cristopher Moore and M. E. J. Newman. Epidemics and percolation in small-world networks. *Phys. Rev. E*, 61:5678–5682, May 2000.

[29] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pages 2431 – 2439, jan. 2002.

[30] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[31] Manoj Parameswaran. Social computing: an overview. *Communications of the Association for Information Systems*, 19:762–780, 2007.

[32] Manoj Parameswaran and Andrew B. Whinston. Research issues in social computing. *Journal of the Association for Information Systems*, 8(6):336–350, 2007.

[33] Douglas A. Perednia and Ace Allen. Telemedicine technology and clinical applications. *The Journal of the American Medical Association*, page 1, 1995.

[34] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 385–388, New York, NY, USA, 2009. ACM.

[35] John Scott. Social network analysis. *Sociology*, 22(1):109–127, 1988.

[36] Adam B. Seligman. *The Problem of Trust*. 1997.

[37] Kisrten P. Smith and Nicholas A. Christakis. Social networks and health. *Annual Review of Sociology*, pages 1–3, 2008.

[38] Kerstin Uvnäs-Moberg. Physiological and endocrine effects of social contact. *Annals of the New York Academy of Sciences*, 807(1):146–163, 1997.

[39] Frank Edward Walter, Stefano Battiston, and Frank Schweitzer. A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems*, 16, 2007.

[40] Fei-Yue Wang, Kathleen M. Carley, Daniel Zeng, and Wenji Mao. Social computing: From social informatics to social intelligence. *Intelligent Systems, IEEE*, 2007.

[41] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.