A Reliable Study of Software Reputation Based on User Experience

- Design a Reliable Software Reputation System with Several Proposed Methods

Xiao Cai

Tie Duan

School of Computing
Blekinge Institute of Technology
Soft Center
SE-37225 RONNEBY
SWEDEN
This thesis is submitted to the Department of Interaction and System Design, School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Authors:
Xiao Cai
E-mail: chandler1999@hotmail.com

Tie Duan
E-mail: duantieren1983@hotmail.com

University advisor:
Yang Liu
School of Computing

School of Computing
Blekinge Institute of Technology
Soft Center
SE-37225 RONNEBY
SWEDEN
4.1.5 User Interface ......................................................... 34
4.1.6 Database .............................................................. 36
4.2 MAIN FUNCTIONS ANALYSIS AND DESIGN .................. 37
4.2.1 Analysis of User Verification .................................... 37
  4.2.1.1 Drawbacks of Restricting Registration by Checking Client’s IP Address ............................................. 37
  4.2.1.2 The Concept of Message Authentication ......................... 38
  4.2.1.3 Message Authentication Code (MAC) .......................... 38
  4.2.1.4 Concept One-Way Hash Function ............................... 39
  4.2.1.5 The Requirements of Hash Function ......................... 41
  4.2.1.6 The Structure of Public-Key Encryption ....................... 42
  4.2.1.7 Applications for Public-Key Cryptosystems .................. 44
  4.2.1.8 Requirements for Public-Key Cryptography ................. 44
  4.2.1.9 Digital Signatures ................................................. 44
  4.2.2 Reading Registry .................................................... 45
4.3 CONCLUSION ............................................................ 46

CHAPTER 5: SCENARIO ANALYSIS .................................. 47

5.1 THE REGISTRATION PROBLEM OF REPUTATION RESULT AND POSSIBLE SOLUTION ................................................. 47
  5.1.1 The Registration Problem of Reputation Result in Existing Systems ......................................................... 47
  5.1.2 Possible Solution in Our Designed System ...................... 47
  5.1.3 Analysis of Scenarios ................................................. 48
5.2 THE PROBLEM OF USER EXPERIENCE AND POSSIBLE SOLUTIONS ................................................................. 48
  5.2.1 The Problem of User Experience in Existing Systems ...... 48
  5.2.2 Possible Solution in Our Design System ......................... 48
  5.2.3 Analysis of Scenarios ................................................. 50
5.3 SCENARIOS OF REPUTATION CALCULATION ........................ 50
  5.3.1 Test Scenarios .......................................................... 50
  5.3.2 Analysis of scenarios .................................................. 54
5.4 OTHER SYSTEMS AND CONTRASTS .................................. 54
  5.4.1 Instance of an Existing System ..................................... 54
  5.4.2 Compared with Other Systems ..................................... 57
5.5 CONCLUSION .............................................................. 58

CHAPTER 6: FUTURE WORK AND CONCLUSIONS .......................... 59

6.1 SELF-CRITICAL DISCUSSION .......................................... 59
6.2 FUTURE WORK OF OUR STUDY ..................................... 59
6.3 DISCUSSION ABOUT DRAWBACKS OF DIGITAL SIGNATURE ........ 59
  6.3.1 Drawbacks of direct digital signatures ......................... 59
  6.3.2 Drawbacks of arbitrated digital signatures ..................... 59
6.4 CONCLUSION OF OUR STUDY .......................................... 60

REFERENCES ............................................................... 61
ABSTRACT

Many users have such experience that after downloading or buying a software product, it does not work well as you anticipated. However, it is lucky to some extent because many other users download malicious software applications from Internet directly. Those problems cause people to think carefully and seriously about reputation of software.

As it is said, the software reputation is very important for users to buy or use software applications, especially for those people who have very few knowledge of computer technologies. Many researches provided various kinds of technical methods to verify whether a software product has good reputation or not. This paper mainly focuses on software reputation research based on user experience.

Nowadays, one effective way to evaluate experience of users is Web Survey, which has increased in popularity because of its convenience, ease of use, low cost, and quickness in knowing the result of a survey. Web survey is widely used to gather user opinions on various purposes. However, most such systems are too weak to keep reliability, and sometimes with a complete fallacious result. After introducing a number of methodologies, we present a concept design of more reliable web survey system based on user experience of software products that are under estimation.

Development tools and some web technologies are discussed, such as PHP and JavaScript that will be used in implementing the web survey system based on concept design of this paper in the future. PHP (Hypertext Preprocessor) is a simple and practical dynamic web page development language, and it is widely used as a multipurpose scripting language embedded in HTML (Hypertext Marked Language). JavaScript is a scripting language and is primarily used in the client-side for development of dynamic websites. Therefore, these two technologies are proper to be applied.

In order to solve the defined research problems, only proposing a method of web survey is not sufficient. The more significant point is reliability of that designed system. Therefore, Digital Signature technology is introduced and described, which will be used for user verification in the system for strengthening reliability of the web survey. The digital signature is widely used in electronic commerce, and it can authenticate one user’s identity effectively. Another important technology that can increase the reliability of comments from users is Reading Registry. As the “Windows” system is the most popular operation system on personal computers today, we choose to study it in this paper. The registry is a basic database of windows system that records every software applications’ information there. So by reading the information in registry, it is easily known whether a user’s comment is valid or not. If the registry records the under estimation software, it means that user indeed used the software and his/her comments are worth consideration, otherwise the comments are too untruthful to be accepted.

Although the web survey system is presented and illustrated by using concept design of this paper, a series of scenarios have been designed to test the real processes of the system. Three main series of scenarios were designed corresponding to digital signature method, registry check method and reputation calculation process respectively. After analyzing the process, those applied methods do strengthen the reliability of the web survey system. Admittedly, there are problems that are not under the control of the system. In the future work, we will carry on studies on them.

Keywords: software reputation, user experience, web survey, reliability
A reliable software reputation report is useful and helpful for normal users to make choices when he/she decide to download or buy a kind of software application. Without which, users may get confused or misguided. Generally, if the application was used well or at least not bad, it can be said that the user made a lucky choice. However, what will happen if the software product is too bad to use after a user buying or downloading it? Moreover, what will happen when it is found that the software product was hardly uninstalled after a user installing it with spending much time? Furthermore, with rapidly increase of Internet around the world today, who can say that the every software product is completely secure after a user downloading it into his/her own computer. So, to answer these questions, a more reliable reputation of software is needed.

Chapter 1 introduces a background of this research paper.

As it is mentioned above, at first some problems of software applications and factors that can influence software reputation are described. Many research papers provided various kinds of technical methods to test or verify software products in order to calculate its reputation result. Nevertheless, before computers can think as human beings, the experiences of users themselves may have the best persuasiveness. Therefore, after that, the paper discusses the user experience and introduces how to capture the user experience. Some effective methods are described, one of which is web survey.

Chapter 2 defines main research problems.

Firstly, the reputation concept of software application should be studied and understood. Therefore, the first research question is how to define the concept of reputation. The section 2.1 “Reputation Concept Description” introduces several reputation reporting systems that existed, and then further describes the reputation concept with two aspects: computer science and economics. Therefore, the answer to first research question is given in this section.

After that, the user experience concept is further introduced. With regard to user experience, what method can be used to capture the user experience is the second defined question. However, to capture the user experience is not the final goal of our study, the purpose of our study is to propose a more reliable reputation system based on user experience. Therefore, the third research question is defined. What method can be used to strengthen reliability of reputation system?

To perform a web survey, a series of criteria for evaluating the quality of software products are needed. The criteria can be taken as guidelines for designing software reputation survey’s questionnaires. After describing the criteria from the document of ISO 9216 and user’s view, the fourth research question is defined that what kinds of information can be used to evaluate software quality? Therefore, to solve these problems occurs in web survey, some techniques are proposed and described in next chapter.

Chapter 3 introduces those applied methodologies.

To do research, the fundamentally applied method is literature review. Without reviewing literature, the basic background knowledge cannot be gained.

Second method is web survey, in the section 1.4 “User experience evaluation methods”, it describes that survey is one of the methods to capture user experience, and of course, the web survey is one method to capture user experience. Therefore, the answer to the second research question is web survey, which is widely adopted because of their convenience, ease to access, ease of use, low cost and real time results. Generally, questionnaires in the survey are in the form
of texts and graphics. Users can click mouse to select answers, which is equally called “make comments” in this paper. However, there are several requirements there. First, each user can and only can make comments to one kind of software product once. The reason is obvious, if one user made comments to a kind of software product repeatedly or he/she continuously changes his/her pseudonyms to make comments, this user can hardly be trusted. Second, the user who makes comments should really use that software application which he/she makes comments. The reason why this requirement is needed is also easy to understand. If the user did not use the software, his/her comments can hardly be trusted and used.

Then, the PHP technology is described as a basic tool for web development. After that, two technologies are introduced: Windows Registry and JavaScript. To access and read registry database, JavaScript technology is one of effective methods that can be applied. Finally, the digital signature technology is introduced. Its definition and some benefits are also described.

More theoretical analysis and explanation regarding technologies of “Registry” and “Digital Signature” will be provided in next chapter. They are crucial methods to solve third defined research problems.

Chapter 4 illustrates a concept design of the web survey system.

System requirements and some needed development tools are firstly illustrated. After that, the system data flow is explained in detail by drawing DFD (data flow diagram). The main user interfaces regarding “survey” and “reputation presentation” are designed by drawing two drafts, one is the survey page and another is reputation presentation page. In survey page, a series of criteria are designed and applied to answer the fourth research question. In order to solve the third research problem, the attentions of the system functions’ analysis are mainly paid to “User verification” and “Reading Registry” functions. These main functions are described particularly which are corresponding to the techniques that described in chapter three.

In the analysis of the function “User verification”, two methods that are IP address checking and Digital Signature are compared. Today, many websites that apply web surveys use the IP address checking function to limit users. In this paper, IP address checking is not recommended because of its inherent drawbacks. Another technique is stricter and better than IP address checking, which is Digital Signature. It has already been used in the domain of network security and is a secure function to authenticate user’s identity. After applying digital signature, the system can make sure that a user only registers once so that one user can make comments to one software product only once at a time.

After verifying the user’s identity, the system still needs to know whether he/she really used the software that was made comments. “Reading Registry” is proposed as another important function to handle this problem. The windows operation system contains every software application’s information at its registry database. So by checking registry, the system can make a decision whether accept this comment for reputation calculation or not. If there are no records about the commented software application, it is most likely that the user is cheating and his/her comments will be discarded.

Chapter 5 designs and analyzes main scenarios and provides a contrast to other systems.

To test the process of the designed system function, a series of scenarios are proposed. They are corresponding to user verification function, reading registry function and reputation calculation function respectively.

After analyzing scenarios, the result of tests indicates that the “User verification” and “Reading Registry” functions are indeed appropriate solutions to strengthen the objectivity of a web survey.
system, and to solve the third research problem. In addition, the reputation calculation function can work correctly.

Chapter 6 describes the future work and conclusions.

In the future work, some important features about key management are discussed. Finally, the conclusions of this study are made.
CHAPTER 1: BACKGROUND

1.1 Reputation problems of software application

Nowadays computers are integrated with daily life increasingly. People store data and share their personal information with them. As the concept of the personal information is implied in this context, the privacy issues will be very important. There are many definitions about privacy but in this work, privacy is ability for individuals to control how personal data are stored or in other word “Privacy is the right to be alone” [1]. In mid 1990s, development of Internet increased and by introducing the web browsers this interest thrived a lot [2]. In this time, companies used Internet to advertise their products that caused much money for them. These kinds of software are the software that now we know them as spyware. The spyware uses the internet connection of users in order to send some information without any permission from them [3]. In other words, they monitor the behaviors of the users without agreements.

As these kinds of software may cause many problems for the system and users, most of them are not familiar to the users [4], the anti-viruses or anti-spywares try to distinguish them, but it is not an easy job and sometimes they cannot be recognized.

Confronting with these kinds of software is a problematic issue since there is no standard or specific model they can base on. However, as most of them invade privacy of the user, we can call them “Privacy-Invasive Software” (PIS) [3, 5]. Most of anti-viruses distinguish the infecting programs, according to their signatures [6], but the point is the legitimacy of software is not consistent among different users. Software that is not legitimate in one’s view can be a useful one for others. On the other hand, in contrary of providing information about installing software in gray boxes under EULA agreement, users are reluctant to read all the parts of this agreement [3, 4, 7, and 8]. For solving this problem, an interesting idea is using other users’ experiences about any specific software products. It will open the concept of reputation systems in this paper. Proof-of-Concept is the name of a tool that is designed based on reputation systems. When a user wants to install or execute a file, this software will ask him/her for their permission to install it. Besides, it may provide the information regarding this software from other users’ experiences. Accordingly, users can share their knowledge and experiences with each other and use their ideas at the same time. In this tool, the classification of different software is based on the level of users’ consent. They define different levels of user’s consent and different levels of severity of negative consequences. The consent of users can change the category of software in this reputation system [9, 4].

1.2 Factors which influence software reputation

Nowadays, the reputation concept is used in several popular websites like www.pconline.cn and www.skycn.com. Those websites do not completely guarantee the honesty of the software uploaded by the promulgators; instead, they provide the users by the information about the reputation of the software. The information of reputation is usually collected by the users’ comments and using experiences. Therefore, it leads us to another problem: some users may provide incorrect and awful comments for a kind of quality software product in order to decrease the reputation of it.

There are different criteria and factors influencing software’s reputation. On the other hand, we have different level of users, which can provide different level of information. The information that an expert provides is more trustable than the information provided by a novice user. In some cases, we have to be cautious of wrong information that some users will provide in order to negate the reputation of software [4]. In addition, users may have different assessments about
specific software and it makes the calculation of reputation more difficult [1, 4], so the concept of reputation depends on different factors.

As a result, the software reputation is influenced by both the real quality of it and the users’ comments. A user’s comments normally should be regarded as the direct feedbacks from his/her using experiences. However, some other factors, like user’s reliability, user’s knowledge of software and so on may also apparently influence the feedbacks of using experiences.

1.3 User experience analysis

In the recent years, the application of different software has increased rapidly. Moreover, ubiquitous computing has become a popular topic in research and design areas. However, the evaluation of pervasive software applications or their influences on users are quite difficult since the evaluation requires analysis of real context. In addition, testing model should have a fully-operational, reliable method so the evaluation with incomplete information will not give a realistic test result. Nevertheless, preliminary tests in early phases of software analysis are necessary to perform in order to achieve information about the end user’s preferences and needs. [10]

The reputation analysis related to software applications with the capturing of user experience has been seen as an important and interesting research issue. In general, user experiences have been captured with techniques like surveys, user comments analyzing and selecting valid information by different reliable methods.

It should be studied that how user experiences can be evaluated in adaptive software applications. User-experience refers to the experience that a person gets when he/she interacts with a kind of software application in particular conditions. In practice, numerous different kinds of people, software applications and environments influence the experience of individual person. The user has the following aspects: values, emotions, expectations and prior experiences, among others. In addition, the software application has influential factors, for instance, security, usability, robust, reliability, compatibility, interaction with users, running speed, resources consumption and so on. All these factors influence the experience of user according to different types of software.

Moreover, to evaluate the reputation of software; there should be methods in order to determine the nature of a product. The type of the software will affect the research methods and targets. Likewise, the evaluation of ubiquitous computing environments emphasizes different factor of software and may thus require different methods for evaluating user experience.

1.4 User experience evaluation methods

There are many kinds of methods of capturing user experiences. For example interviews, diary, surveys, observation, prototyping and storytelling [11]. The surveys, storytelling and diary can be applied for long-term usage to gets information from user experience [12]. This is the result from that users are able to effectively record their usage experiences. Stories can organize and remember experiences, so they let users communicate with each other in some different scenarios. However, nonverbal expressions of user are important sometimes because users might not be conscious of their own experiences. Observation method can deal with that. Buchenau and Fulton [13] developed Experience Prototyping for simulating experiences at different scenarios. Designers, clients and users may experience themselves than only observing other users’ experience.

Ubiquitous environments offer new aspects to user experience research. User experience in such type of challenging environment and system has been assessed by interviews and observations. Bellotti et al. [14] utilized several different ways in the evaluations. In the first one, two different
questionnaires: complete version and short version respectively. A year later, they performed ethnographic observation, with qualitative and quantitative measurements. Johanson et al. [15] have developed interactive workspaces and performed a number of experiments for HCI (Human Computer Interaction). They used open meetings with different sets of participators in their experiments. Fleck et al. [16] developed an electronic guidebook for an interactive museum, namely Exploration. Some unconventional studies of users were performed which observe users with and without technologies in that museum.
CHAPTER 2: PROBLEMS DEFINITION

In this chapter, four research problems will be defined and described. Section 2.1 focuses on reputation concept and gives some introductions with regard to computer science and economics fields. Section 2.2 describes the user experience concept and capture method for it. Section 2.3 introduces the criteria of software product’s quality, which is also an essential point for software reputation.

2.1 Reputation Concept Description

Nowadays, in virtual communities the reputation systems have already frequently used. Those systems strengthen reliability among users, whether their purposes are to expand practice of auctions or to increase applications of software reputation. To study the software reputation system, we firstly need to understand the basic definition or definitions of reputation. Therefore, the first research problem is defined that what the concept of reputation is. Therefore, this chapter starts with the review of the previous research about reputation in this chapter.

2.1.1 Reputation Reporting System

In electronic commerce, there is a so-called reputation reporting system. Such system has been already implemented and applied. Several research reports have shown that seller reputation has significant influences on on-line auction prices, especially for high-valued items [17].

eBay, which is a so famous online website for shopping has a function of the accumulative positive and negative ratings for a seller or buyer over a recent period (a week, a month, or a year). This function provided in eBay is a practice instance of reputation reporting system. Resnick and Zeckhauser have empirically analyzed this reputation system and concluded that the system may encourage transactions [18]. Probably, if the buyer wins, he/she will remit the payment as promised; for the seller, probably he/she will send the auctioned products once the payment is received. The economic analysis indicates that reputation has important effects on price. Both Lucking-Reily et al., Bajari and Hortacsu have empirically examined coin auctions in eBay [19] [20]. Moreover, such economic research have tested and verified that human’s experience indeed affects the reputation results in internet auctions.

In present models, some conceptual gaps still exist. Resnick and Zeckhauser have pointed out a particular effect of the eBay reputation reporting system [18] in their study, that refers to the positive and negative feedbacks from users. The positive feedbacks are obviously disproportionate and the negative feedbacks are rarely seen. Without valid feedbacks, the rational choice can hardly be made by using those reputation report systems.

In addition, those studies did not consider the effects of cheat. Therefore, this kind of reputation reporting system is too easy to be attacked by malicious users. Significantly, the arbitrary change of online pseudonym did not sufficiently attract those researchers.

2.1.2 Computer Science

In electronic commerce, reputation generally plays an important role in distributed systems. The reputation system in the anonymous storage system is used to create an accountability system for users [21]. Trust management in the system allows users to publish materials anonymously such that censorship of and tampering with any publication in the system is rendered very difficult [22].
In computer science literature, Marsh [23] is one of the first to introduce a computational model for trust in the distributed artificial intelligence (DAI) community. However, as he pointed out, a few limitations occur in his simple model. Firstly, in the model, trust is represented as a subjective number ranges from minus one to positive one. The model reveals problems at the extreme values and at zero. Secondly, operators and algebra for manipulating the trust values are limited, and troubles emerge when the model is dealing with negative trust values. The difficulties about the concept of “negative trust” and its propagation are also pointed out by Marsh.

Abdul-Rahman, et al., has studied reputation as a form of social control in the context of trust propagation-reputation, which is used to influence agents to cooperate for fear of gaining a bad reputation [24]. They have considered that the reputation is a propagated notion. The reputation effect is passed to others by means of word of mouth.

Sabater, et al. has defined reputation as the “opinion or view of one about something” and have modeled three notions of reputation: individual, social, and ontological [25]. Individual reputation is focused on the topic of how an individual’s opinions are judged by others. Social reputation is focused on the opinions of individuals based on the reputation of different social groups they belong. Ontological reputation refers to the multidimensional nature of reputation depending on particular contexts.

Yu, et al., has proposed probabilistic models for reputation. Reputation for an agent is inferred based on propagated ratings from an evaluating agent’s neighbors. These propagated ratings are in turn weighted by the reputation of the neighbors themselves. [26]

2.1.3 Economics

Economists have widely studied the reputation issues. With the increasingly development of electronic game, the reputation in game theoretic settings is paid more attention. Many economic studies on reputation have relationships to repeated games. In many current online games, reputation of players is considerably important for cooperative tasks or balances. Game theorists have assumed the existence of such balance since the 1950’s in the so-called Folk Theorem [27]. Economists often interpret the sustenance of cooperation between two players as evidence of “reputation effects” [28].

The game theorists often study the entry deterrence by using reputation notions. Kreps and Wilson referred Harsanyi’s theory of imperfect information about players’ payoffs to explain “reputation effects” for multi-stage games [29] [30]. Their studies reveal that an incumbent company has the motives to receive an early reputation for being “tough”, in order to decrease the probability for future entries into this industry. More recently, Tadelis has studied reputation at the firm level — firm reputation being a function of the reputation of the individual employees [31].

2.2 User Experience

With the rapid increment of software applications, more and more people interact with the various software products everyday. Therefore, the evaluation of user experience has become a necessary research topic. The second research problem is what method can be used to capture the user experience? However, to capture the user experience is not the final goal of our study, the purpose of our study is to propose a more reliable reputation system based on user experience. Therefore, what method can be used to strengthen reliability of reputation system is defined as our third research question.

First, we have to understand the meaning of user experience. Various definitions of user experience have been already proposed. Cawthon, N. et al. argues that user experience is “a
subject commonly tied to interactive applications—typically software and web interfaces which holistically describes the relationship a user has when using an application and the resulting product of this interaction” [32]. Goto, K. defines user experience as “the overall perception and comprehensive interaction an individual has with a company, service or product” [33]. In addition, in Wikipedia, user experience is defined as “a term used to describe the overall experience and satisfaction a user has when using a product or system” [34].

From these definitions, we can easily draw a conclusion that user experience is something people feel before, during and after they have interacted with a system, a service and a software application; or user experience is to some extent an emotional state. However, considering that emotion is something that rather difficult to capture and evaluate, in the section we would not concentrate on the emotional aspect, but rather on user’s experience and feedback. Experience consists of a user’s academic, technical, cultural, historical and aesthetic acquisitions. For example, the user experience mainly refers to the experience that comprises the academic and technical acquisitions when he/she is using a software product.

2.3 Important Criteria for Evaluating the Quality of Software

Today, software products have been widely used by common users who are dealing with varied types of software applications everyday. However, not all of those software products can fulfill the users’ different requirements. Therefore, the fourth research problem focuses on what kinds of information can be used to evaluate software quality.

2.3.1 ISO 9126 Criteria

Specifying the quality of a specific software product is now difficult for users or developers. The user needs to clearly understand and successfully communicate his/her requirements for some software product that is to be developed. The developer needs to thoroughly understand the requirements from users and confidently make sure whether it is possible to provide the product with the required software quality.

ISO 9126 can be used to decrease misunderstanding between user and developer [35]. It provides the definitions of the characteristics and evaluation process of software products’ quality. Six quality characteristics are defined, which are intended to be exhaustive. The standards described in ISO 9126 could be criteria for evaluating the quality of software products. Applied criteria could prevent repeating work in case of the software product did not meet the user’s requirements.

The six characteristics are listed:

**Functionality** is the set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs. [35]

**Reliability** is the set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period. [35]

**Usability** is the set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users. [35]

**Efficiency** is the set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions. [35]

**Maintainability** is the set of attributes that bear on the effort needed to make specified modifications. [35]
Portability is the set of attributes that bear on the ability of software to be transferred from one environment. [35]

2.3.2 Users’ View

Common users are mainly interested in the usability, the performance and the effects of the software products. Common users evaluate a product without knowing the internal mechanisms and the development of the software product. Therefore, from a common user’s point of view, he/she may just ask some general questions to make sure whether the software product is good or not.

Users’ questions may include:

• Are all the required functions available in the software product?
• How reliable is the software product?
• How efficient is the software product?
• Is the software product easy to use?
• How easy is it to transfer the software product into another environment?

2.4 Conclusion

This chapter proposes and defines four research problems:

1. What is the concept of reputation?
2. What method can be used to capture the user experience?
3. What method can be used to strengthen reliability of reputation system?
4. What kinds of information can be used to evaluate software quality?

After problems definition, the next process is to find methods to handle or solve these research problems. The answers of first and second questions have already provided in this chapter when they are proposed and described. However, further information and methodologies on basis of the third and fourth research problems will be illustrated in the following chapters.
CHAPTER 3: METHODOLOGY

This chapter will mainly describe the methodologies of the research paper. A primary flow chart (Figure 3.1) is illustrated to clarify our basic flow of our research work. At first, literature review is used to understand and analyze the basic knowledge of this research paper. After gaining sufficient knowledge and information, we form and describe the research problems and proposed four questions to our study.

Then, we continue to collect data for seeking effective solutions to our questions. The main data collection type we used is documents. We choose both public and private documents in order to get enough data. Those important findings regarding our designed software reputation system are introduced in this chapter, like web survey, PHP, JavaScript, Windows Registry and Digital Signature. Web survey is the core and foundational method of our designed system. In our designed software reputation system, we will use web survey to capture user’s experience in order to calculate the reputation. The software reputation system is designed to be a web-based system. PHP and JavaScript technologies are introduced and recommended as development tools to implement the system. The collected data of registry and digital signature technologies are particularly described in this chapter since they are chosen to handle our research problems. They will be placed in our designed system as two main functions, which can obviously strengthen the reliability of software reputation system. In addition, during the data collection process, we observe the activity of an instance code of JavaScript that can read registry data in windows system.

After collecting sufficient data, we plan to design a reliable software reputation system in the chapter 4. The data flow diagram of the system will be displayed to illustrate the modules and data flow of that designed software reputation system. User interfaces and database tables will also be designed. Furthermore, we plan to analyze the two crucial functions of system that apply the methods “Digital signature” and “Read registry” we studied in data collection.

To test our design, in chapter 5, we will design several scenarios of currently existed issues in most of software reputation systems, such as pseudo usernames and fake comments. By comparing the behaviors of our designed system in the scenarios with an unreliable software reputation system, we will test that whether our designed system theoretically strengthen the reliability of the software reputation system or not.

At last, we plan to discuss some weakness of our research works to our study in chapter 6. In the future, we plan to continue to refine and implement our design and test it in practice.
3.1 Literature Review

The literature review in a research study shares with readers the results of other studies of that are closely related to the study being reported [36]. It can fill the gaps between previous study and ongoing research, and establish a basic background for comparing with the other researcher's study results.

A literature review for a proposal or a research study means that locating and summarizing the studies on one topic. Often these summaries can be recognized as research studies. To conduct a scholarly literature review, we applied a series of steps:

Begin by identifying useful key words in locating materials in an academic library at universities or colleges. With these key words in our brain, then begin searching the library or some online databases catalog for holding (for example, journals or books). Some important computerized
databases are used and helpful for literature reviews, like IEEE, ACM, and Google Scholar. They are available online. Initially, approximate fifty reports of research in articles or books related to research on topic are tried to locate. Then, we set a priority on the search for journal articles and books because they are easy to locate and acquire. After quickly skim the literatures, the literatures are classified into different fields. Therefore, in the study, we can correctly link them to our research problems. At last, the literature review ends with a summary of major themes found in the literature and we continue to perform further research work on the topic along the lines of the proposed study.

3.2 Data Collection

We choose both public and private documents in order to get sufficient and useful data. The important findings regarding our designed software reputation system are introduced in this section, like web survey, PHP, JavaScript, Windows Registry and Digital Signature. In addition, in section 3.2.5, we observe the activity of an instance code of JavaScript.

3.2.1 Web Survey

As it is said in section 1.4, there are several kinds of methods have been used to capture user experiences. For example interviews, diary, surveys, observation, prototyping and storytelling. “Surveys” is an effective method to capture user experiences.

“Surveys” is a traditional method for effectively collecting information or feedbacks from people. As the applications on Internet are so widespread today, the development of surveys on the World Wide Web is also rather rapid. It is believed that soon Internet surveys will replace those traditional methods of survey. Certainly, others still argue that whether the web survey will be able to play the primary role in the survey’s industry. However, what is clear now is we are confronting a new time of survey industry, though how the trend will evolve is not yet completely defined. No matter how the web survey research will evolve in the future, the current method of web survey is worth applying as an original tool to capture or collect information.

Each coin has two sides; web survey is also the same. In one side, the data collection of web survey could be extremely numerous. Thanks to web survey, the common people around the world can easily make their own surveys on websites, putting questionnaires there and collecting data. In addition, for those researchers, the cost for getting access to many specific databases of dissertation, periodical or literature is dramatically lower than traditional methods of survey. Even a normal website can launch a large-scale data collection in our society today. The large-scale survey is no longer a particular activity, which only governments or some big organizations can perform before. Because of the comparatively low cost of web survey, it is possible that every person access the website and potentially democratizing the process of survey. Furthermore, it is feasible for web survey to contain multimedia content that will attract more people with different backgrounds. The web survey could be regarded as a very new world of survey industry, some features of which are extremely difficult to accomplish by using traditional methods.

In another side, the web survey has its potential risks. With the rapidly increment of web survey, it will become more and more difficult to distinguish the good surveys from those bad ones. People may intentionally change the value of web surveys, so the result of web survey is limited and to some extent is hardly trusted. Well-designed, high-quality web surveys may be very well overwhelmed by the mass of other data-collecting activities on the Internet. In summary, then, while web surveys normally may become increasingly easy to do (both cheaper and quicker), good Web surveys (as measured by accepted indicators of survey quality) may become hard to carry out [37].
After describing the basic concept and characteristics of web survey, we need to discuss the process and the elements of a web survey.

There are four primary parts of a web survey’s operation. The first is questionnaire. Questionnaires in the survey are commonly in the form of texts and graphics, and there are some multimedia questionnaires in many websites today. Respondents then can write words or click mouse to answer questions. Such kind of method is quick, accurate and easy.

The second part we need to introduce is database, which stores all the information of survey in a website. Normally, the respondents enter the data directly into the database, and then the collected data is sent into a calculation module within minutes.

Calculation module is the third part of a web survey. It is in charge of calculating the data read from database and sending the result data of processing back to database again.

The last part of a web survey is presentation. Presenting the result can be considered as the ultimate goal of a survey. In a web survey, the presentation is performed by web pages that are more quick and efficient than traditional methods.

Therefore, we could use these basic parts to perform a web survey. Firstly, respondents read the questionnaires displayed on a web page. Then, they could answer the questions by typing or clicking mouse; meanwhile, their comments are directly sent to database and stored. Thirdly, the calculation module gets the data from database and starts to process. The results of calculation are sent back to database again, and calculation module finishes until a new task comes. Finally, the result of survey is displayed on web pages, which is so-called presentation.

As we mentioned above, a good web survey may become hard to carry out. To develop a web survey system, its usability should be first thing that needs concern. Usability mainly referred to “user friendly” [38]. The term usability was replaced with the term “quality in use” [39] in most recent fashion. Therefore, this concept is important to the designing of the web surveys and becomes the crucial factor to accomplish the web site design that bases on organized objectives. There are several major guidelines (see Table 3.1) describe the core elements of usability, which is concise to understand.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Learnability-time to learn</td>
<td>Learnability</td>
<td>Learnability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learnability-retention</td>
<td>Memorability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Effectiveness-task time</td>
<td>Efficiency</td>
<td>Efficiency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Effectiveness-errors</td>
<td>Errors</td>
<td>Effectiveness</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Operability</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Understandability</td>
<td></td>
</tr>
</tbody>
</table>

| User view (Subjective)       | Flexibility  | Satisfaction | Satisfaction | Attractiveness |

Table 3.1 The usability core elements from major guidelines [40]
Besides usability, there is another key factor of web survey: reliability. Respondents not only just answer the questions but also need to see the reliable result. If the web survey cannot try to keep reliability, everything that has already done for a web survey would not make any sense.

### 3.2.2 PHP Technology

A web survey is based on the web application system. To study the web survey method, the web development technologies should be considered at the beginning.

PHP (Hypertext Preprocessor) technology is a widely used scripting language. It is especially suited for web development and can be embedded in HTML (Hypertext Marked Language). HTML is a simple marked language to write hypertext documents, which are so-called web pages stored in websites. Generally, PHP runs on a web server, taking PHP code as its input and creating web pages as its output. It works free on almost every web server and platform, and it can be used with most relational database management systems. PHP is installed on more than 20 million websites and 1 million web servers. [41]

PHP is originally designed for producing dynamic web pages. In 1995, PHP was created by Rasmus Lerdorf. However, now The PHP Group produces the main implementations of PHP and serves as the de facto standard because there is no formal specification. PHP is a free software product that is released under the PHP License; however, such license is not compatible with the GNU General Public License (GPL).

#### 3.2.2.1 Usage

PHP takes input from a file or stream and outputs another stream of data. The stream may contain text and/or PHP instructions. Generally, the output of PHP is HTML file. In addition, it automatically detects the language from users.

The primary focus of PHP is scripting on server side, and it is similar to other server side scripting languages, such as ASP (Active Server Pages), Sun Microsystems’ JavaServer Pages, and mod_Perl. A number of frameworks are designed and applied for PHP, which progresses RAD (Rapid Application Development). Those frameworks offer building blocks and design structures that are easily used. There are, for example, CakePHP, Symfony, Code Igniter, Zend Framework and so on.

In 2004, PHP 5 was released. This new version completes the object models, applies Zend 2 engine (the second generation of Zend Engine), improves the design of the grammar and reinforces the connection with MySQL (a popular relational database management system). The release of PHP 5 is indeed a milestone, since then, PHP becomes a real object oriented and powerful scripting language.

Until April 2007, PHP has already installed on over 20 million servers on Internet, and it has recorded as one of the most popular modules of Apache. Many famous websites are partly written or completely written in PHP, such as Facebook, Wikipedia, Yahoo!, MyYearbook, Digg, Word Press, and Tagged etc.

In addition, PHP can be used to create stand-alone applications and for shell scripting.

#### 3.2.2.2 Syntax

PHP parses code by its delimiters. It will send everything outside the delimiters directly to the output without parse. Most usual delimiters are `<?php` and `?>`, which mean open and close
respectively. Delimiters `<script language="php">` and `</script>` are also valid. Two short tags are used to start code, `<? or `<?= (echo a string or variable) and one tag to end code, `?>`. We can use these delimiters to separate the PHP code from the other code, including HTML code.

The next important element of PHP’s syntax is variables that are prefixed by a dollar mark. The interesting thing is the type of a variable does not need to be defined at first. Variables are case sensitive, which is contrary to function and class. To use a string, we can quote the values with double-quoted mark. PHP serves a new line as white space in terms of a free-form language (new lines inside string quotes are exceptional). A statement is ended by a semicolon. For the types of comments syntax, `/* */` are treated as block comments; `//` and `#` are treaded as inline comments. PHP provides several ways to output text, one of which is the famous statement echo.

PHP is similar to many high-level languages that follow the C syntax style in terms of keywords and language syntax. The keywords `If`, `for` and `while` represent conditions and loops respectively. Syntax about functions is also similar to the syntax in C language.

```
<HTML>
<HEAD>
<TITLE> PHP Test </TITLE>
</HEAD>
<BODY>
  <?php echo '<p>Hello World</p>'; ?></BODY>
</HTML>
// PHP code embedded in HTML [41]
```

### 3.2.2.3 PHP Function Instance

“gethostbyname()” is a library function in PHP. It gets the IP address corresponding to a given host name (PHP 3, PHP 4, PHP 5) and returns the IP address of the host specified by `hostname` or a string containing the unmodified `hostname` on failure.

Format of this function:

```
string gethostbyname ( string hostname )
```

A simple `gethostbyname()` example:

```
<?php
$ip = gethostbyname('www.example.com');
echo $ip;
?>
```

### 3.2.3 Windows Registry

Since the windows operation system has became the main choice of general customers around the world, the registry database of this operation system is chosen as an instant for illustrating.

The Registry is a database that stores settings and information of installed software in Microsoft Windows operating systems. It contains useful data for all the hardware, OS (operating system), general software and users’ settings. The registry also provides a window into the operation of the kernel, exposing runtime information such as performance counters and currently active hardware. [42]
The first registry occurs in Windows 3.1, since then, it becomes a famous technological term for the windows family. Its intention is to arrange effectively the large quantity of INI files that previously were used to record settings for windows programs. These files are scattered all over the system and are hardly managed.

### 3.2.3.1 Keys and Values

The registry contains two basic elements: keys and values.

The keys of registry are similar to folders. Each key can have sub keys of it, which may have further sub keys and so on. The syntax of keys is similar to system’s pathname. Backslashes are used to indicate the level of hierarchy. For instance, `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows` refers to the sub key "Windows" of the sub key "Microsoft" of the sub key "Software" of the HKEY_LOCAL_MACHINE key. [42]

The values of registry are name or data pairs that are stored in keys. They are separately referred to keys. The backslashes are also may be used in values names, which makes them hard to distinguish from their key paths. Some windows APIs (Application Program Interface) are used to query and manipulate registry values. Those APIs can take values’ names separately from their key paths or identify their parent keys.

In Windows 3, the term is a holdout of the 16-bit registry, in which keys cannot contain any arbitrary name or data pairs. However, it contains only an unnamed value (should be a string). In this way, the whole registry is more like an associative array, where the keys form a kind of hierarchy, and values are all strings. When Microsoft created 32-bit registry later, they confused to some extent the additional capabilities of creating multiple named values per key and the meanings of the names.

Here are a number of different types of the Registry values:

<table>
<thead>
<tr>
<th>REG_NONE</th>
<th>No type</th>
</tr>
</thead>
<tbody>
<tr>
<td>REG_SZ</td>
<td>A string value</td>
</tr>
<tr>
<td>REG_EXPAND_SZ</td>
<td>An “expandable” string value that can contain enviroment variables</td>
</tr>
<tr>
<td>REG_BINARY</td>
<td>Binary data (any arbitrary data)</td>
</tr>
<tr>
<td>REG_DWORD/REG_DWORD_LITTLE_ENDIAN</td>
<td>A DWORD value, a 32-bit unsigned integer (numbers between 0 and 4294967295) (little-endian)</td>
</tr>
<tr>
<td>REG_DWORD_BIG_ENDIAN</td>
<td>A DWORD value, a 32-bit unsigned integer (numbers between 0 and 4294967295) (big-endian)</td>
</tr>
<tr>
<td>REG_LINK</td>
<td>Symbolic link (UNICODE)</td>
</tr>
<tr>
<td>REG_MULTI_SZ</td>
<td>A multi-string value, which is an array of unique strings</td>
</tr>
<tr>
<td>REG_RESOURCE_LIST</td>
<td>Resource list</td>
</tr>
<tr>
<td>REG_FULL_RESOURCE_DESCRIPTOR</td>
<td>Resource descriptor</td>
</tr>
</tbody>
</table>
REG_RESOURCE_REQUIREMENTS_LIST | Resource Requirements list
---|---
REG_QWORD/REG_QWORD_LITTLE_ENDIAN | A QWORD value, a 64-bit integer (either big or little-endian, or unspecified) (Introduced in Windows 2000)

Table 3.2 A list of registry value types[42]

### 3.2.3.2 Hives

The Registry is split into some logical sections, or so-called "hives". Hives are in general given such name by the windows API definitions, which all begin with "HKEY". They are further abbreviated to a three or four short names beginning with "HK" (for example, HKCU or HKLM).

“The HKEY_LOCAL_MACHINE” and “HKEY_CURRENT_USER” nodes have a similar structure. Typically, software applications check the settings for them in "HKEY_CURRENT_USER\Software\Vendor's name\Application's name\Version\Setting name" at first to query data. If the settings are not found, the applications look instead at the same location under the key “HKEY_LOCAL_MACHINE”. When writing settings back, the applications use the contrary method — “HKEY_LOCAL_MACHINE” is written first. If the settings cannot be written back (this usually happens if the user is not an administrator), the data of settings are stored in the key “HKEY_CURRENT_USER instead”. [42]

Here is an instance of Hives described below:

The key is “HKEY_LOCAL_MACHINE”, and it is abbreviated to HKLM. HKEY_LOCAL_MACHINE stores the general settings of all users on the computer. On NT-based versions of windows systems, HKLM contains four sub keys: SAM, SECURITY, SOFTWARE and SYSTEM. In the folder “%SystemRoot%\System32\config”, their respective files locate there. The fifth sub key is HARDWARE that is unstable and dynamical. Consequently, such key is not stored in files. The information about hardware drivers and services of the computer are located under the sub key SYSTEM, while the sub key SOFTWARE records the settings of all general software applications and windows programs.

### 3.2.4 JavaScript Introduction

"JavaScript" is a brand of Sun Microsystems Company, and it hardly has any relationships with the famous program language, Java. One of the reasons for using JavaScript language is that, it is a widely used scripting language in today’s web applications. Almost every browser supports JavaScript nowadays. It is primarily used in the form of client-side scripting language for the development of dynamic websites. [43]

Originally, JavaScript is based on the standard of ECMAScript. It is characterized as a dynamic and weakly typed language with first-class functions. JavaScript was affected by many other languages and was designed to look like Java. However, JavaScript is much easier than high-level languages for any programming beginners to work with.

Despite its name, JavaScript is not related to the Java language even if they do have a few similarities. Both of them use syntaxes influenced by the classic syntax of C language. JavaScript borrows many names and naming conventions from Java language. This name is the result of an economic deal between two companies of Netscape and Sun,

Because of its simplicity, the specific grammars and operations are not discussed in our paper, an introduction of JavaScript is enough here.
3.2.5 Instance of Using JavaScript to Read Registry

It is sufficient for us to know that using JavaScript can readily read the values of windows registry. In fact, this process is essentially performed by particular windows APIs that are called by JavaScript at running time.

The “WshShell.RegRead()” method returns the value of a key or value-name from the windows registry.

```javascript
var WshShell = WScript.CreateObject("WScript.Shell");

WshShell.RegWrite("HKCU\Software\ACME\FortuneTeller\", 1, "REG_BINARY");
WshShell.RegWrite("HKCU\Software\ACME\FortuneTeller\MindReader", "Goocher!", "REG_SZ");

var bKey = WshShell.RegRead("HKCU\Software\ACME\FortuneTeller\MindReader");
WScript.Echo(WshShell.RegRead("HKCU\Software\ACME\FortuneTeller\MindReader"));

WshShell.RegDelete("HKCU\Software\ACME\FortuneTeller\MindReaderi");
WshShell.RegDelete("HKCU\Software\ACME\FortuneTeller\"));
WshShell.RegDelete("HKCU\Software\ACME\"));
```

3.2.6 Digital Signature Introduction

The purpose of using digital signature is because it can be considered as a kind of “finger print” in order to verify the user’s identity and it could effectively ensure the security and protect personal privacy during data or information transferring.

The scheme of digital signature is a type of asymmetric cryptography. When messages transfer in an unreliable channel, the receiver needs to believe the messages were indeed sent by the claimed sender. In such circumstance, a proper digital signature can be implemented to offer the authentication.

Digital signatures are similar to traditional handwritten signatures in many ways; in fact, it is more difficult to forge a proper digital signature than a handwritten one. However, a proper digital signature is based on the schemes of asymmetric cryptography (by using private and public keys) and must be correctly implemented. Digital signature can also provide non-repudiation, which means the signer cannot successfully claim they did not sign a message. Furthermore, some non-repudiation schemes provide a time stamp for the digital signature; therefore, even though the encryption is no longer valid, the signature is still valid. Messages may be anything represented as a bit-string: examples include electronic mail, contracts, or a message sent via some other cryptographic protocol. [45]

Digital signature is often used to implement electronic signatures. A broader term may refer to any electronic data that needs to signature. However, not all electronic signatures use digital signatures. For example, in the United States the European Union, electronic signatures have legal significance.

The basic procedure of digital signature can be described by three phases. At fist, the sender make a summary of the message by using a particular method (hash function, we will illustrate in detail in chapter 4). Then he/she encrypts the message and sends both the ciphertext and the summary to receiver. When getting the ciphertext and the summary, the receiver sets about to decrypt the ciphertext into original message, and make a summary of it by using the same method. Finally, the receiver can compare the received summary with the summary just generated. If these two summaries are the same, it means the sender is successfully authenticated. The summaries here are considered as the signatures for that message. More details about the technologies are illustrated in chapter 4.
3.2.6.1 Definition

A digital signature scheme typically consists of three basic and key algorithms:

1. A key generation algorithm and it selects a private key uniformly from a set of possible private keys at random. The algorithm outputs the private key and a corresponding public key.

2. A signing algorithm which, given a message and a private key, produces a signature.

3. A signature verifying algorithm which given a message, public key and a signature, either accepts or rejects.

Two main properties are required. First, a signature generated from a fixed message and fixed private key should verify on that message and the corresponding public key. Secondly, it should be computationally infeasible to generate a valid signature for a party who does not possess the private key. [45]

3.2.6.2 Instance of Benefits of Digital Signature

Digital signatures can be used to authenticate the source of the transferred messages. If the owner of a digital signature secret key is bound to a particular user, a valid signature can authenticated that the message was sent by that claimed user. The importance of such authentication is especially obvious in a financial context. For example, assume a bank's branch sends instructions to the headquarters requesting a modification to the balance of an account. If the headquarters cannot be convinced that such a message is truly sent from an authorized source, acting on such a request could be a big mistake. [45]

3.3 Conclusion

Chapter 3 described the methodologies of our study work. A flow chart of research work is displayed to visualize those steps. Literature review has been used at the beginning of the study, and the problems definition is described in chapter 2. Therefore, the data collection is described in detail in this chapter. The content of data collection here is more focus on findings rather than method introduction.

After study those findings from data collection, we inferred a theory: The proposed methods “Digital signature” and “Registry check” could strengthen the objectivity and reliability of a software reputation system. These methods are also the solutions to our research questions.
CHAPTER 4: A DESIGN OF RELIABLE SOFTWARE REPUTATION SYSTEM

In this chapter, we are going to introduce our ideas of a reliable software reputation system. First, we are going to talk about the system requirements and explain the system functions with system data flow diagrams. The system features are further explained and analyzed in system data flow part according to their interactions with each other.

Secondly, we will propose the key technologies designed in our system including “Registry Check” and “Digital Signature”. We point out why they are important and how they are implemented to examine the users’ comments and verify users’ identities.

4.1 Software Reputation System Design

4.1.1 System Requirements

The Reputation system is designed to collect user feedbacks or comments from the user's point of view. So first, there are several important requirements should be illustrated.

1. Users should really use the software that the reputation system needs to assess. If the users did not use or experience the software themselves, their feedbacks or comments are hardly believed.
2. Web survey pages of the system should be clear and easily understood. To make reputation system more trustful, more feedbacks or comments from users needed to be collected. Correspondingly, survey pages should be designed clearly and not so complicated; otherwise users may feel too bored to leave his/her experience.
3. Each user may only give his/her feedbacks or comments once to the under-evaluation software product. This requirement is important to avoid cheating.
4. The system should interact with users to provide the information of software that need to be evaluated and results of reputation calculation.
5. The system should calculate the reputation of software products correctly. If the reputation data of a kind of software product changes, the system can updates the reputation results correspondingly.
6. Administration functions needs to be considered. As an administrator of reputation system, he/she should be able to manage users’ data, survey pages, and comments to system and so on.

4.1.2 Development Tools

The reputation system will mainly use four widely used web technologies. First, we mentioned in chapter three is PHP (Hypertext Preprocessor) script language. PHP is a script language used to provide dynamic HTML documents in server-side.

Second is JavaScript and HTML (Hyper Text Mark-up Language). JavaScript is a scripting language that used to enable programmatic access to objects in applications and is primarily used in the form of client-side. Dynamic websites are using JavaScript widely and frequently nowadays. HTML is the predominant markup language for web development. It provides a method to describe the basic structure of text-based information in a document. It denotes certain text as links, headings, paragraphs, lists, etc and supplements those texts with interactive forms, embedded images.
The third is Apache HTTP Server, which is commonly referred to simply as Apache. It is a web server notable for playing a key role in the initial growth of the World Wide Web and became the first web server nowadays.

The last is a relational database management system, named MySQL. MySQL runs as a server providing multi-user access to the database. Its popularity for use with web applications is closely tied to the popularity of PHP.

4.1.3 System Flow

The new user will be presented a register page first, and get a digital signature application for client-side to register himself/herself. Then, as a registered user, the system will verify the user’s identity to decide whether pass this user or not. After successful user verification, a page that contains lists of software products’ information will be shown to the user who then can search and choose one software he/she wants to make comments. Next, the web survey page of the chosen software application will be presented to user. He/She then can give his/her comments based on using experience. During such process, the system will check the client’s registry to make sure whether the user’s comments are valid or not. After checking, reputation results will be calculated and presented, so users can see the reputation calculation results of every software product recorded in the system.

If someone is not the registered user of the system, he/she can also see the reputation result page of each kind of software product. It means that the reputation system provides software reputation information for everyone who browses this web page; but not everyone can arbitrarily modify the reputation results.

4.1.3.1 Data Flow Diagram

After the basic flow of the system was described, data flow diagram (DFD) can be shown. A data flow diagram is a graphical representation of “data flow” through a software system. DFD can also visualize the data processing when a structured system design is performing.

Figure 4.1 is primary DFD at level 1, and the following diagrams are decomposed from above one. First, four basic elements of data flow diagram are displayed:

<table>
<thead>
<tr>
<th>Function</th>
<th>Input / Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database / File</td>
<td>Data flow</td>
</tr>
</tbody>
</table>

Table 4.1 Elements of data flow diagram

“Function” is some thing can transform or process the input data into outputs.

“Input/output” represents the object that generates input data or may accept output data.

“Database” notation here is used to present a kind of database table or file which stored in the system. They can be indexed by F1, F2, and F3, so these files are bale to be easily addressed and used. In relational database management system, the database file can be also considered as a two-dimensional table. More details about database tables will be described in section 4.1.6.

“Dataflow” which is a line with an arrow or arrows, represents a flow of data with the direction that is indicated by the arrow or arrows.
At level 1, the whole reputation system is considered as an entire entity first. There are three outside objects around it: Unverified user, Verified user and Administrator respectively.

Unverified user is the user who did not be verified by system. For new users who plan to register, the system will get users’ information and let him/she download the digital signature (SS) client. So new users can use SS client to send digital signature data for system verification, and the result of verification will be sent back; for those users just want to browse reputation pages, system gets their browsing requests and then presents the reputation pages to them.

If users are registered and verified, they can send requests, which may contain requests’ information for particular page, or his/her comments for some software to system. Therefore, system will process those data and provide reputation results directly or after calculation.

The third object is administrator, who is designed to in charge of the supplement of software products. If system finds out that a software product is requested but not existed in database, it will send a notice to inform administrator that one software product lacks, so administrator can come and handle this.

In addition, administer is able to be responsible for daily maintenance; however those operations are not needed parts of data flow diagram here.
In Figure 4.2, the entity of “Reputation system” was decomposed into three main functions: function1 “User login process”, function2 “Reputation process” and function3 “Software collection”.

Function1 is designed to handle the users’ information. Before a user can make comments to software in system, his/her information must pass function1 to distinguish different kinds of users, make records for new users and verify registered users. Database table F1 and F2 are needed. After function1 successfully verified a user, function2 “Reputation process” will be noticed and started to acquire verified user’s request data to handle, during that procedure, database table F1, F3, F4, F5 are used. Details of these two functions and tables they used will be illustrated at level3.

Function3, “Software collection” interacts mainly with administrator. F3 (Software hold table) contains the existed software’s information, and F4 (Software lack table) records the software
which was requested but not existed in system. Function 3 tells administrator which software product lacks, after administrator supplementing, the record of supplemental software will be added to F3 for future use.
Figure 4.3 DFD at Level 3
In Figure 4.3, the function 1 of level 2 “User login process” is further decomposed into 1.1 “User control”, 1.2 “New user record”, 1.3 “User verify” and 1.4 “Reputation presentation” functions at level 3.

When an unverified user comes, his/her information or browsing request will firstly reach function 1.1 “User control”. Function 1.1 distinguishes users’ request: whether he/she just wants to browse pages, whether he/she is a new user who wants to register or whether he/she has already registered and needs to login. Mark “⊕” represents the meaning of “OR”. Therefore, if the user just wants to browse pages, the request data will be sent to function 1.4. If there is a new user and wants to register, a digital signature client will be offered for downloading and the user information will be sent to function 1.2 and if it is a registered user who wants to login system, his/her data will be sent to function 1.3 to verify.

Function 1.4 “Reputation process” is used to send back the reputation result page to users. It gets the browsing request data and sends the page back to user. During this procedure, database table F3 (Software hold table) and F5 (Reputation table) are needed. F3 provides existed software’s information and F5 records the reputation results. The data from F3 and F5 will be queried and combined by function 1.4 the page of reputation result will be generated and sent.

Function 1.2 “New user record” is used to handle new user’s register. After it gets the new user’s information, the function will record them into F1 (Users table) and F2 (Keys table).

Function 1.3 “User verify” can verify the user’s identity by using digital signature. As we mentioned, each registered user will download a digital signature client, so a ciphertext will be sent as registered user’s information to function 1.3. The user’s information and keys will queried and read from F1 and F2 for verification. After that, a result will sent back to user and if verify successfully, a notice which tells this user was verified will be sent to function 2.1 for further work.

Function 2 of level 2 “Reputation process” is further decomposed into 2.1 “Reputation presentation”, 2.2 “Registry check”, and 2.3 “Reputation calculation” respectively. When a user passed the verification, he/she is a verified user, which means he/she can send their requests and comments with higher privilege to system. Word “request” can be used to represent the data from user.

Function 2.1 “Request handling” is in charge of distinguishing different kinds of requests. If a verified user just wants to check the result of reputation, his request will be sent directly to function 2.4 for handling. If he/she makes comments to a software, firstly function 2.1 queries and reads F3 to present the requested software, if a requested software is not existed, the lack information will sent to F4, otherwise their comments will be sent to function 2.2 for check.

Function 2.2 is named “Registry check”, which is another important method to keep the reliability of system besides digital signature. If a user made his/her comments to software, system needs to know whether the user really used or is using the software application. The function gets the formulated comments from function 2.1 and after successful checking, a valid comment will be sent to function 2.3. If incorrect result gets, the function will add a record to F1.

Once a valid comment comes, the function 2.3 will process the data and calculate the reputation for the software. F5 (Reputation table) which contains different kinds software’s reputation values is needed. This function queries and reads the records from F5 to calculate the reputation result of the software product by using both existed data in database and new data from comments. After that, it stores the new reputation into F5 of the software and sends a notice to function 2.4 to inform reputation records are updated.
Function 2.4 “Reputation presentation” gets the requests of users or the notice of function 2.3 to present the reputation of the software by querying and reading both F3 and F5. As a result, the user can acquire the reliable software reputation results based on user experience that is captured by web surveys.
4.1.4 Primary System Architecture

Figure 4.4 demonstrates the primary overview architecture of this web survey system. MySQL will be utilized as the database management system. The database contains the users’ data, software’s data, and reputation evaluation data. The Apache web server embedded with the PHP engine generates dynamically pages upon users’ requests.

![System Architecture Diagram](image)

4.1.5 User Interface

We left out general routine pages such as registration, administrative pages and demonstrate page. As seen in Figure 4.5, the draft of survey page allows the user to rate the reputation of software “BitComet” according to a series of criteria. We designed several important criteria based on the discussion in section 2.3:

- Usability (The quality of the software in use)
- Executing Speed (Is the running speed tolerable to users.)
- Efficiency (How much it consumes the system resource?)
- Security (Whether it invades user’s privacy or is it easy to uninstall?)

These criteria are important information and guidelines for evaluating the quality of software products; therefore, they can also be regarded as a solution to the fourth research problem.

At first, users can read a general instruction of a certain software product, and then he/she could start the web survey by pressing the “Recommend Button”.

In Figure 4.6, another draft of the reputation presentation page of software “BitComet” is shown. There are four basic reputation results shown to users, which are relative to the criteria on survey, so the user can get multidimensional reputation results of a software product. The four bars with different colors are marked with some special meanings. Color “Red” means the reputation is too low which is contrary to color “Blue”, and the rest color represents a moderate state. In addition,
the percentage numbers are also presented at the end of each colorful bar in order to give more accurate reputation results. The words “out of 100 verified users” at the lower right corner of the page are used to tell the users the sum of valid surveys at this moment.

**BitComet:**
BitComet is a fast and easy-to-use Bit Torrent/HTTP/FTP download client. It is P2P file-sharing freeware and one of the most popular P2P protocols designed for high-speed distribution. BitComet supports simultaneous download, DHT networks (trackerless), a download queue, selected downloads in the Torrent package, fast-resume, disk caching, speed limits, auto port mapping, proxies, and IP filtering.
Version 1.11 has elevation is not needed to associate torrent file in Windows Vista and save to My Document in HTTP batch download dialog is supported.

**Make a comment for (BitComet):**

**Usability (The quality of the software in use)**

**Executing Speed (Is the running speed tolerable to users?)**

**Efficiency (How much it consumes the system resource?)**

**Security (Whether it invades user’s privacy or is it easy to uninstall?)**

Figure 4.5 draft of survey page
4.1.6 Database

Relationship database are composed by relationships between the tables and data tables themselves. A data table is commonly considered as a kind of two-dimensional table, which has rows and lines. Each table illustrates some special objects and their attributes, or some aspects of the database. Therefore, a row commonly represents a kind of characteristics and a line represents a record.

In this system, there are five database tables are initially designed. See Figure 4.7.

Users table contains information of a user, for example: User ID, name, age, credibility and so on.

Table Keys, are designed particularly to record public keys of users and private key of it. Digital signature technology will perform depend on it. The line between Users and Keys tables is relationship of these two tables, which means that one user can hold one public key from system.

Software_Hold table holds the content of software. System can use this information to give users an introduction of this software.
Software_Lack table that contains the information of software products that are not recorded in system can be used to let administrator manage and upgrade the database better. This table contains only the software lack information whereas the other tables contain data of those existed software products. It means the records of this table cannot be found in other tables, therefore there is not a key can be found to connect this table with others either.

Reputation table records survey results from calculation function of system according to the criteria of a software product, therefore system can also present the reputation result according to this table. In addition, a relationship is needed between Reputation and Software_Hold tables.

In all database tables, PK stands for primary key and FK stands for foreign key. Database tables use keys to connect each other.

### Figure 4.7 Database Tables

**F1 Users table**

<table>
<thead>
<tr>
<th>PK</th>
<th>User ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>User name</td>
<td>Age</td>
</tr>
<tr>
<td>Profession</td>
<td>Credibility</td>
</tr>
</tbody>
</table>

**F2 Keys table**

<table>
<thead>
<tr>
<th>PK,FK1</th>
<th>User ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public key</td>
<td>Private key</td>
</tr>
</tbody>
</table>

**F3 Software hold table**

<table>
<thead>
<tr>
<th>PK</th>
<th>Soft_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soft_name</td>
<td>Version</td>
</tr>
<tr>
<td>Type</td>
<td>Characteristic</td>
</tr>
<tr>
<td>Publisher</td>
<td>Reputation_ID</td>
</tr>
</tbody>
</table>

**F4 Software lack table**

<table>
<thead>
<tr>
<th>PK</th>
<th>Lack_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soft_name</td>
<td>Version</td>
</tr>
<tr>
<td>Type</td>
<td>Publisher</td>
</tr>
</tbody>
</table>

**F5 Reputation table**

<table>
<thead>
<tr>
<th>PK</th>
<th>Soft_ID</th>
<th>Reputation_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>Exe_Speed</td>
<td></td>
</tr>
<tr>
<td>Efficiency</td>
<td>Security</td>
<td></td>
</tr>
</tbody>
</table>

### 4.2 Main Functions Analysis and Design

#### 4.2.1 Analysis of User Verification

#### 4.2.1.1 Drawbacks of Restricting Registration by Checking Client’s IP Address

In Web Survey, some websites are using voting system for providing the reputation of software applications; this is used by gathering and evaluating user comments. Those websites do not guarantee the honesty of the software uploaded by the promulgators completely; instead, they provide the users by the information about the reputation of the software.
However, there exists one possibility that is that many user accounts can be registered by one IP address. Therefore, this causes the problem: one user may register several user accounts to increase or decrease a software reputation.

One of the solutions is to restrict one user from registering several usernames by checking his/her IP address.

However, there is one possible drawback of this solution: a user may change one’s IP address in order to escape the restriction by checking or locating one’s IP address. For instance, a user may change his/her network environment or log into a different network, and then he/she gets a different IP address. Since the solution above cannot totally prevent this problem.

So now, we are going to discuss the usage of digital signature and how could it keep the unique identity of certain user.

4.2.1.2 The Concept of Message Authentication

Encryption is discussed here because it protects different types of attacks including active attacks and active attacks. For example, eavesdropping is a passive attack and falsification of data and transactions is an active attack. Therefore, message authentication is known as a concept of protection against those attacks.

Collections of data, document and file could be considered as message. If a message is from its alleged source and been verified, it can be considered as an authentic message. The procedure about communicating parties verifying their received message is said to be message authentication [46].

4.2.1.3 Message Authentication Code (MAC)

One technique of authentication includes the usage of a secret key and it is used to generate a small block of data. This block of data is appended to the message and said to be message authentication code. It assumes that two communicating parties A and B, a secret key $K_{AB}$ are shared between them.

If A wants to send a message to B, the message authentication code is calculated as a function of message and key: $MAC_M = F(K_{AB}, M)$. It transmits plus code of the message to the target recipient. The same calculation on the received message is performed by the recipient. In addition, the calculation is performed with the same secret key in order to generate a new message authentication code [47].
The receiver calculates the message with the same algorithm, and then the result code is compared with the received code (Figure 4.8). We assume that the secret key is only kept between the receiver and the sender, in addition, the received code and the calculated code is matched, we can conclude that:

First, the message has not been altered and the receiver is verified. We assume that attackers do not know the secret key and it is impossible for them to alter the code to alterations in the message correspondingly. Therefore, if the message has been changed by an attacker, the result code calculated by the receiver will differ from the received code.

Second, since the secret key is kept in secret and no one knows it, nobody could fake a message with a correspond code. Therefore, the message from its alleged sender could be verified by the receiver.

Last, we assume that a sequence is included in the message, and then the receiver can be verified of the sequence since the sequence number cannot be changed by attackers.

4.2.1.4 Concept One-Way Hash Function

One-way hash function is another algorithm related to message authentication code. However, there are differences between them. A variable-size message \( M \) can be accepted by a hash function as input. Correspondingly, a fixed-size message digest \( H(M) \) will be calculated as output. This is not like message authentication; a secret key is not taken as input by a hash function. The digest will be sent with the message in order to authenticate a message [46] [47].

The message could be authenticated in three ways and they are illustrated in Figure 4.9. Part a shows that conventional encryption could encrypt the message digest. We assume that the encryption key is only kept between the sender and receiver, then authenticity is assured.

Part b shows that public-key could also encrypt the message. Public-key will be furthered discussed in Section 4.2.1.8. There are two advantages of using public-key: first, a digital
signature will be provided as well as message authentication. Second, the distribution of keys to communicating parties is not required by public-key.

(a) The usage of conventional encryption

(b) The usage of public-key encryption
A hash function is used with no encryption for message authentication is shown in part c.

We assume that a secret value $S_{AB}$ is shared between two communicating parties A and B. If A wants to send a message to B, the hash function will be calculated through the message and the secret value: $MD_M = H(S_{AB} || M)$.

A then sends $[M || MD_M]$ to B. B will calculate $H(S_{AB} || M)$ again with $S_{AB}$ and compare the result with $MD_M$. If an attacker intercept the message, it is impossible for him/her to modify it since he/she does not know the secret value. Therefore, attackers cannot fake a message as long as the secret value is kept in secret.

### 4.2.1.5 The Requirements of Hash Function

A hash function could provide a “finger print” for a message or file, therefore, it is important in message authentication and digital signatures. A series of requirements are listed below about hash function $H$:

1. Any size of a block of data could apply $H$.
2. A fixed-length output will be produced by $H$.
3. Any given $x$ could be easily computed by $H(x)$. This could make hardware and software implementations practical.
4. Computationally, it is impossible to find $x$ such that $H(x) = h$ for any given value $h$. It is referred as the one-way property.
5. Computationally impossible to find $y \neq x$ with $H(y) = H(x)$ for any given block $x$. It is referred as weak collision resistance.
6. Computationally, It is impossible to find any pair $(x, y)$ such that $H(x) = H(y)$. It is referred as strong collision resistance.

The practical application of a hash function to message authentication requires the first three properties. The “one-way” property of the fourth property is easy to provide a code given a message, however, actually infeasible to produce a message given a code. The property is necessary if we assume that a secret value (Figure 4.9 part c) is included in the authentication technique [46].

The secret value will not be sent and if we assume the hash function is not one-way, the secret value could be discovered by an attacker. In addition, if a transmission could be intercepted and...
the message $M$ and the hash code $C = H(S_{AB} \| M)$ could be obtained by an attacker. Since the attacker now knows $M$ and $S_{AB} \| M$, the hash function could be inverted and the attacker could obtain $S_{AB} \| M = H^{-1}(C)$, it is a trivial matter to recover $S_{AB}$.

In the fifth requirement, it is infeasible to find any message with the same hash value as a given message. If an encrypted hash code is performed (Figure 4.9a and b), a forgery could be prevented. We assume that this property is not true, an attacker could intercept a message with encrypted hash code, then he/she could calculate an unencrypted hash code and an alternate message with the same hash code [46] [48].

A weak hash function satisfies the first five requirements. It is a strong hash function if satisfies the sixth requirement.

4.2.1.6 The Structure of Public-Key Encryption

Diffie and Hellman proposed public-key encryption in 1976. The algorithms of public-key algorithms are based on mathematical function. In addition, the cryptography is asymmetric with the performance of two separate keys. It is different from symmetric encryption that only uses one key [46].

A series of components of public-key encryption scheme are listed below (Figure 4.10a):

1. **Plaintext**: The plaintext is considered a readable message or data and it will be calculated into the algorithm as input.

2. **Encryption algorithm**: Various transformations on the plaintext are performed by the encryption algorithm.

3. **Public and private key**: Public and private key are considered as a pair of keys that have been selected. One of them is used for encryption; the other is used for decryption. The encryption algorithm performs he exact transformations preformed by using e public or private key as input.

4. **Cipher text**: Cipher text is considered as scrambled or unreadable message; it is produced as output from the plaintext and the key. Different two keys will generate two cipher texts for a given message.

5. **Decryption algorithm**: The original plaintext will be calculated from cipher text by this algorithm.
The public is made known to public for others to use; the private key is kept in secret from its owner. One key for encryption and the other for decryption is a general purpose of public-key cryptographic algorithm [46].

The following steps of public-key authentication are necessary:

1. Every user has a pair of key in order to encrypt or decrypt his/her message.

2. One of the two keys is placed in public register. It is said to be the public key. The other key is kept private. As shown in Figure 4.10a, each user maintains a collection of public keys obtained from others.
3. For instance, Bob uses Alice’s public key to encrypt the message if he wants to send a message to Alice.

4. Alice will decrypt the message by using her private after she receives Bob’s message. No one else could decrypt the message since the Alice’s private key is only known by Alice.

Therefore, all participants are able to access to public keys; private keys are kept private by each participant and should never be distributed. If a user changes his/her private key, his/her, old public key should also be replaced by the new one calculated with the new private key.

### 4.2.1.7 Applications for Public-Key Cryptosystems

In public-key system, the sender could choose to use his/her private key or the receivers’ public to implement cryptographic functions depending on the application. Three categories of the use of public-key cryptosystems are listed below [46] [48]:

1. Encryption or decryption: The sender uses the receiver’s public key to encrypt a message.

2. Digital signature: A message is “signed” by its sender with the private key. This procedure is achieved by a cryptographic algorithm applied to the message that is a function of the message.

3. Key exchange: The exchange of a session key is cooperated by two sides.

### 4.2.1.8 Requirements for Public-Key Cryptography

A series of requirements of public-key cryptograph are listed below:

1. Computationally, it is easy for a party B to calculate a pair of keys including public key $\text{PU}_b$ and private key $\text{PR}_b$.

2. Computationally, it is easy for a sender A to calculate the corresponding cipher text $C = E(\text{PU}_b, M)$ with the public key and the message $M$ to be encrypted.

3. Computationally, the receiver B could easily use the private key to decrypt the resulting ciphertext and recover the original message: $M = D(\text{PR}_b, C) = D[\text{PR}_b, E(\text{PU}_b, M)]$

4. Computationally, it is impossible for an opponent to determine the private key, $\text{PR}_b$ if he/she knows the public $\text{PU}_b$.

5. Computationally, it is impossible for an opponent to recover the original message $M$ if he/she knows the public key $\text{PU}_b$ and a cipher text $C$.

6. For a pair of two keys, either key can be used for encryption, with the other used for decryption: $M = D[\text{PU}_b, E(\text{PR}_b, M)] = D[\text{PR}_b, E(\text{PU}_b, M)]$

The sixth requirement is not mandatory for all public-key applications. [46]

### 4.2.1.9 Digital Signatures

Public-key encryption can be used in another way, as shown in Figure 4.10b. We assume that in a web site system, the client side wished to send a message to the server side. It does not matter
that the message is kept in secret or not, however, the server side wants to make sure that the message is indeed from the client side.

In this case, the client encrypts the message by using his/her private key. After the server receives the cipher text, if it can be decrypted with the client’s public key, this proves that the message must come from the client side. Since nobody knows the client’s private key, therefore, no one could fake a cipher text that could be decrypted by the client’s public key. A digital signature is served by the encrypted message. Nobody could change the message without client’s private key. Therefore, the message is authenticated in source and integrity.

Storage is required for validating author and contents and a copy should be kept in cipher text so that the origin and contents could be verified in case of a dispute. However, encrypting a small block of bits, which is a function of the file. This is a more efficient method to achieve the same result. The block of data is said to be authenticator, it should be impossible to change the document without changing the authenticator. A signature will be served to verify origin, content, and sequencing if we assume that the authenticator is encrypted with the sender’s private key.

The confidentiality is described but not provided by the encryption. This means that the message is not safe from eavesdropping since the signature is based on part of the message and the rest is transmitted in the clear. [46]

4.2.2 Reading Registry

In order to increase the objectivity of software reputation evaluation, reliable comments from user are definitely needed. In web survey, user comments are important to be selected according to user experiences. Therefore, it leads us to a problem that some users may provide incorrect and awful comments for some qualified software in order to decrease the reputation of them.

One possible solution can be used to examine user comments by checking installation or execution records stored in client’s operation system (Windows Registry) under user's agreement.

Many websites are built with JavaScript programming language for client side, so we use JavaScript code as an instance to illustrate how to check user’s registry:

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>New Document</title>
<meta name="Generator" content="EditPlus">
<meta name="Author" content="">
<meta name="Keywords" content="">
<meta name="Description" content="">
</head>

<script language="javascript">
function mediaPlayerInstallLocation()
{
var obj = new ActiveXObject("WScript.Shell");
var s="HKLM\SOFTWARE\Microsoft\MediaPlayer\Installation Directory";
var sNic = obj.RegRead(s);
alert(sNic);
}
</script>
```
4.3 Conclusion

After designing the software reputation system, the system structure is completed and the system functions are integrated together for further testing.

We have explained the system diagrams and tested the two technologies (digital signature verification and registry check). We are going to test the system features by designing scenarios in next chapter.
CHAPTER 5: SCENARIO ANALYSIS

In this chapter, we are going to describe several scenarios in order to test the system features (see chapter 4) and compare with other existing systems. After analyzing the result of scenarios, we will explain how they solve the problems based on our research methods.

Then we will introduce an example of existing system and compare the system features with our designed system. We will also discuss the advantages and disadvantages of both two systems in our conclusion part.

5.1 The Registration Problem of Reputation Result and Possible Solution

5.1.1 The Registration Problem of Reputation Result in Existing Systems

In existing reputation systems, if a new user wants to use the system functions, he/she needs to register a new user account with username and password in the system. Then the user is able to view the information of software and rate the software or add his/her comments to it. The process will be explained by the following two steps

Step1:
We suppose that there is software called “Adobe 9.0” in the system, and there are 100 persons have rated it and 50 of them recommended it. As a result, the reputation of “Adobe 9.0” is 50%.

Step2:
Now there is a new user who is going to rate it, and he recommends it by voting for it. The reputation of the software will be \(\frac{50+1}{100+1}\) which is 50.5%

We assume that each user has one chance to vote for a certain kind of software. Now there is a possibility that a single user may register several accounts and then he/she is able to vote for the software more than once.

In Step 2, if the user votes for “Adobe 9.0” the second time, the software reputation will be \(\frac{50+2}{100+2}\) which is 50.9%. If he/she votes for it the third time, the reputation result will be \(\frac{50+3}{100+3}\) which 51.4%

For this reason, if the user registers 100 times and then he/she is able to vote for the software 100 times and the reputation result will be \(\frac{50+100}{100+100}\) which is 75%

This is a malicious behavior of user, which decrease or increase the software reputation in purpose. It causes the unreliable reputation of the software and misleads other user’s choice to certain kind of software.

In addition, if the IP address of user is restricted with the registration, the user could change his/her IP address to register repeatedly.

5.1.2 Possible Solution in Our Designed System

The solution will be tested by the following scenario:

A new user wants to login to the system. First, he/she needs to register to the system, so the system will ask him/her to offer basis information like username, password, address and personal number. In addition, system will offer the user a client setup application of digital signature for
downloading. After the user download and execute it, he/she will complete the user registration process with digital signature verification. The digital signature verification process will be executed every time when a user wants to login to the system.

This solution restricts that one user is only allowed to have one account. It protects the software reputation result against malicious behavior, which decrease or increase the result in purpose. The digital signature application will be executed every time when a user login to the system. In addition, if a user does not want to download the client application, he/she will also be able to check the information of software and reputation rated by other users, however, he/she is not able to rate software himself/herself since he/she does not pass the digital signature verification.

5.1.3 Analysis of Scenarios

Those test scenarios were created so that it enabled testing the features of the system. Moreover, it made it possible to evaluate user experience in the real environment. The study illustrated that the system functions of digital signature verification are appropriate methods for evaluating user and verify user’s identity in user-system interaction.

The designed system functions not only prevent repeated registration from decreasing the reputation of certain software but also enable the convenience for unverified user to view web pages and obtain information of software.

5.2 The Problem of User Experience and Possible Solutions

5.2.1 The Problem of User Experience in Existing Systems

In existing reputation systems, a user could rate the software according to his/her experience. User experience is something people feel before, during and after they have interacted with a system, a service and a software application. For instance, a user is professional editor, he has used the software “Microsoft Office 2003” before; he considered that this software is not good enough to be recommended in the system. Therefore, he will choose to vote against to it, and this activity will decrease the reputation of “Microsoft Office 2003”.

Now we will face another problem: how to confirm this user has indeed experience to the software. For instance, this user did not use this software before or he even does not know it at all, he just wants to decrease its reputation in purpose. What can we do to protect against this behavior and how can we identify these kinds of issues before they take place?

5.2.2 Possible Solution in Our Design System

According to the system third level data flow diagram (see Figure 4.3), after a verified user made comments to a software, system will check whether he/she used to apply the software or not. If the user did not apply the software before, we consider that his/her comments are not recommended for reputation calculating. Therefore, the function of checking registry in the client’s side is performed. After a user made comments to software, the registry check function will perform. If there are records of the software in user’s operation system registry, the comments from this user will be used in calculating reputation, if not, his/her comments are not valid and the system will make a record in database.

The solution will be tested by the following scenarios:
A verified user “Alice” answers the questions of survey page and then leaves his/her comments of the software “Adobe Acrobat 9.0”.

Then the system begins to perform and access the client’s operation system registry to find out if the corresponding software information is stored there. For example (see Figure 5.1):

In the Registry tree structure, the HKEY_LOCAL_MACHINE key’s sub key SOFTWARE stores every software application’s information that installed in the client’s computer. The first value under sub key SOFTWARE is “Adobe” and it contains the information of software “Acrobat 9.0” in the computer.

The comments of “Alice” about software “Adobe Acrobat 9.0” are valid and the system will continue to process her comments.

Table 5.1 Scenario of valid comments

![Figure 5.1 Registry Tree Diagram](image)
A verified user “Bob” answers the questions of survey page and then leaves his comments for the software “Microsoft Office 2003”.

Then the system begins to perform and access the client’s operation system registry to find out if the corresponding software information is stored there. For example (see Figure 5.1):

There is no such value “Microsoft Office 2003” under sub key SOFTWARE, which may contain the information of software “Microsoft Office 2003” in the computer.

The comments of “Bob” about software “Microsoft Office 2003” are invalid and will be discarded by system, meanwhile “Bob” will also be recorded for his unreliability this time.

<table>
<thead>
<tr>
<th>Invalid comments scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>A verified user “Bob” answers the questions of survey page and then leaves his comments for the software “Microsoft Office 2003”. Then the system begins to perform and access the client’s operation system registry to find out if the corresponding software information is stored there. For example (see Figure 5.1): There is no such value “Microsoft Office 2003” under sub key SOFTWARE, which may contain the information of software “Microsoft Office 2003” in the computer. The comments of “Bob” about software “Microsoft Office 2003” are invalid and will be discarded by system, meanwhile “Bob” will also be recorded for his unreliability this time.</td>
</tr>
</tbody>
</table>

Table 5.2 Scenario of invalid comments

5.2.3 Analysis of Scenarios

In these scenarios, we assume that if the registry check method was not applied, the system will accept all kinds of comments of survey from all verified users. Users who have not used that kind of software can also make his/her comments to system for reputation calculation. The problem is, those comments without indeed user experiences of the software are unreliable, and this unreliability will obviously cause the inaccuracy in the system. Therefore, a reputation system based on web survey is not completely reliable if its users can make comments to software arbitrarily without any reliability checks.

As we discussed above, the advantages of registry check are that it can make the user’s comments more reliable and keep the system more objective. This method can check whether the users applied the software or not. This designed system function is a suitable method to keep objectivity of web survey system.

5.3 Scenarios of Reputation Calculation

5.3.1 Test Scenarios

The reputation results are formatted to percentage numbers in the system, which is like a kind of voting to some extent. For example, one software application’s reputation number of “Executing speed” is 80% and its reputation number of “Usability” is 60%. That means 80 percent verified users recommend “Executing speed” of this software and 60 percent users consider “Usability” as a recommendation. Correspondingly, the number also means that 20 percent users do not think the “Executing speed” of this software is recommended and 40 percent users think that “Usability” of it is not good.

Test scenarios:
User “Alice” made comments to the software “BitComet” and there is no reputation result of “BitComet” in the system now. Therefore, the system will record this software and initialize its reputation value.

The comment of “Alice”:
Usability ---------------------- Recommend (add 1)
Executing Speed ----------------- Recommend (add 1)
Efficiency ---------------------- Not Recommend (do nothing)
Security ---------------------- Not Recommend (do nothing)

So the system will calculate the new reputation (1 user):
Usability ---------------------- [(1)/1] ------------------ 100.00%
Executing Speed ----------------- [(1)/1] ------------------ 100.00%
Efficiency ---------------------- [(0)/101] ------------------ 0.00%
Security ---------------------- [(0)/101] ------------------ 0.00%

The results have been rounded up or down and formatted as percentage.

After calculation, the results will be sent to database and shown on the reputation presentation page (see Figure 5.2). Alice will see the reputation results she just made, and other users can see the reputation results of the software “BitComet”.

<table>
<thead>
<tr>
<th>Initial reputation calculation scenario</th>
</tr>
</thead>
</table>

Table 5.3 Scenario of initial reputation calculation
**BitComet:**

BitComet is a fast and easy-to-use BitTorrent/HTTP/FTP download client. It is P2P file-sharing freeware and one of the most popular P2P protocols designed for high-speed distribution. BitComet supports simultaneous download, DHT networks (trackerless), a download queue, selected downloads in the Torrent package, fast-resume, disk caching, speed limits, auto port mapping, proxies, and IP filtering. Version 1.11 has elevation is not needed to associate torrent file in Windows Vista and save to My Document in HTTP batch download dialog is supported.

<table>
<thead>
<tr>
<th>Reputation: (BitComet)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Usability</strong></td>
</tr>
<tr>
<td><strong>Executing Speed</strong></td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
</tr>
<tr>
<td><strong>Security</strong></td>
</tr>
</tbody>
</table>

Table 5.2 Scenario of initial reputation calculation
User “Bob” made comments to the software “BitComet”, and reputation result of “BitComet” has already existed. For example:

- **Usability**: 65.00%
- **Executing Speed**: 72.00%
- **Efficiency**: 58.00%
- **Security**: 22.00%

The new comment of “Bob” are (100 users):

- **Usability**: Recommend (add 1)
- **Executing Speed**: Recommend (add 1)
- **Efficiency**: Not Recommend (do nothing)
- **Security**: Not Recommend (do nothing)

Then the system will calculate the new reputation (101 users):

- **Usability**: \( \frac{(0.65\times100+1)}{101} \approx 65.36\% \)
- **Executing Speed**: \( \frac{(0.72\times100+1)}{101} \approx 72.28\% \)
- **Efficiency**: \( \frac{(0.58\times100)}{101} \approx 57.43\% \)
- **Security**: \( \frac{(0.22\times100)}{101} \approx 21.78\% \)

The results have been rounded up or down and formatted as percentage.

After calculation, the results will be sent to database and shown on the reputation presentation page (see Figure 5.3). Bob will see the reputation results, and other users can see reputation results of the software “BitComet”.

### Table 5.4 Scenario of reputation update

---

53
5.3.2 Analysis of scenarios

Those test scenarios were created in order to test the function “Reputation calculation” of the system. In addition, they simulate possible situations of calculating reputation in real environment. The scenarios have shown that the reputation calculation function is an appropriate method for calculating the reputation results of software products.

5.4 Other Systems and Contrasts

5.4.1 Instance of an Existing System

Nowadays there are many web sites systems, which offer all kinds of information about software and its reputation. Here is an instance of web site www.softs.com [50]:

It offers users by information and downloads about different kinds of multimedia converters, applications to different video/DVD, audio converter, and video converter to Mac and Windows, security software, browsers and so on.
Figure 5.4 is the customers’ registration in this web site; new customer account is registered by Email address. After registration, customers will be contacted and receive the latest software information sent by Email.

Figure 5.4 User login [51]

In addition, the web site has two kinds of memberships: common user memberships and author memberships. Author is more like a developer who has the option to submit software. The differences are listed in Figure 5.5 below:

Figure 5.5 Two Types of user [52]

Once user login to the website, he/she is now able to view the information about software, see Figure 5.6, we use Firefox browser as an example:
Publisher's description of Mozilla Firefox 3.0.9

From Mozilla:

Mozilla Firefox is a fast, full-featured Web browser. Firefox includes pop-up blocking, tab-browsing, integrated Google search, simplified privacy controls, a streamlined browser window that shows you more of the page than any other browser and a number of additional features that work with you to help you get the most out of your time online.

Version 3.0.9 fixes the issue where a corrupt local database caused Firefox to "lose" its stored cookies and fixes several security issues and stability issues.

Figure 5.6 an instance of Software [53]

There are general information of the software and video introduction as well for users to read and watch. Then users are able to check the features of the software such like download, rating and so on in Figure 5.7 below:
5.4.2 Compared with Other Systems

Comparing our designed system with other system, we have listed the different features in table 5.5 below:

<table>
<thead>
<tr>
<th>System Features</th>
<th>Designed Reputation System</th>
<th>Other Systems</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Login</td>
<td>✓</td>
<td>✓</td>
<td>For user to login and view information or add user's comments</td>
</tr>
<tr>
<td>Software Information</td>
<td>✓</td>
<td>✓</td>
<td>General information or introduction to certain kind of software</td>
</tr>
<tr>
<td>Rating Software</td>
<td>✓</td>
<td>✓</td>
<td>For user to rate software and the calculation result will effect the software reputation</td>
</tr>
<tr>
<td>Author Login</td>
<td>x</td>
<td>✓</td>
<td>For author memberships to login</td>
</tr>
<tr>
<td>Author Options</td>
<td>x</td>
<td>✓</td>
<td>For author to submit software</td>
</tr>
</tbody>
</table>
### Verification Features

<table>
<thead>
<tr>
<th>User Verification</th>
<th>Digital Signature Verification</th>
<th>Verifying the identity of user</th>
</tr>
</thead>
<tbody>
<tr>
<td>√</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>User Experience Verification</td>
<td>Registry Verification</td>
<td>Make sure user has real experience to certain software</td>
</tr>
<tr>
<td>√</td>
<td>×</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.5

#### 5.5 Conclusion

In the chapter, we have designed several scenarios to simulate possible problems in real environment. Through analysis to scenarios, we identified that which kind of information is important to be verified in our system. We can conclude that the reputation system is capable to verify identities of users by implementing the digital signature verification. In addition, the registry check feature could ensure the objectivity of user experience about certain kind of software, based on hypotheses scenarios and research methods, it could effectively protect the reputation calculation result against malicious behaviors which decrease or increase software reputation in purpose.

The system functions could restrict the user registration, confirm indeed user experiences; it protects the reputation result against malicious behaviors of user. Therefore, the software reputation will be kept reliably and objectively in our designed system.
CHAPTER 6: FUTURE WORK AND CONCLUSIONS

6.1 Self-Critical Discussion

We have tested system functions by designing scenarios. Compared to other existing systems, there are still several disputed points need to be further discussed:

Firstly, the reputation system could offer a client application for digital signature verification. The argued point is if a user does not want to download it. Therefore, the user will not be able to rate software in our system. However, he/she still could view web pages of basic software information and check the reputation result rated by other users.

Secondly, the designed system only supports users who are using Window platform. For instance if a user is using Linux operating system, he/she may only view web pages but not register in the system with digital signature verification. This is we are going to study further more in future.

6.2 Future Work of Our Study

1. Continue to develop the reliable software reputation system based on our study.

2. Study other technologies that can strengthen the reliability of system. For example if the client’s browser does not support JavaScript, we need some other methods to access and read user’s registry. One suggested method is to use a C/C++ program as a client-side application. After user accepting and installing it, this application will directly send the related data of user’s computer to server.

3. Study other operation systems, for example Unix/Linux.

6.3 Discussion about Drawbacks of Digital Signature

There are two types of digital signatures: direct digital signatures and arbitrated digital signatures. In order to understand both of them, it is important to know the differences of them.

6.3.1 Drawbacks of direct digital signatures

First, we are going to talk about direct digital signatures. A message sender contacts the message receiver and the sender’s public key will be given to the receiver. A secure message will be sent to the receiver by the sender, then the receiver decrypt the message and read the contents by using the sender’s public key.

It seems that this method is more secured than involving a third party; however, some drawbacks are existed related to it. For instance, a sender could claim that keys were compromised and he/she could deny sending any message. In addition, the security of the information is only equal to the private key’s security of the sender. If the digital key is compromised, information of message is possible to be sent with the compromised key.

6.3.2 Drawbacks of arbitrated digital signatures
The other type of digital signatures is arbitrated digital signature. We assume that the message sent by the sender is not compromised. A third party validates the identities of the sender and receiver between message transfers. One of the drawbacks of it is that a third party must be involved. In validating entities and contents of messages, the trusted third party should be an active role. Therefore, it may cause a bottleneck in message traffic. However, the problems in direct digital signatures seem to be solved in arbitrated digital signatures [54].

6.4 Conclusion of Our Study

By defining the reputation concept in software analysis, we were able to design our Software Reputation System. We have proposed two techniques including Digital Signature and Registry Check for the user verification part in our system. We have shown that since digital signature can be considered as a secure “finger print” for identifying certain user and useful in protecting privacy, it can be used as a reliable function in the system for verifying users’ identities. In addition, registry check function is effective to distinguish whether a user actually has the experience to certain kind of software.

The test scenarios have demonstrated the features of our designed software reputation system, the hypotheses of problems in real environment have been tested and solved based on our research methods.

Through all scenarios, the possible situations have been tested and functions of the system have been verified. As a result based on hypothetical scenarios, the reputation system is capable to verify identities of users and effectively protect against malicious behaviors that decrease or increase the software reputation in purpose.

In our designed system, we have proposed the methods of using digital signature verification and registry check. Our system focuses on improving the user verification process, but through comparing with other current systems, there are still some drawbacks of our system: We did not design the author or developer membership in our system; it is a special kind of user. As a developer, one can uploads software and this behavior not only offers more information and available software to customers but also presents experiences that are more professional from a developer. To some extent, it increases interactions between all kinds of users and provides more patterns of user communications.

We pointed out major drawbacks about digital signature technology and they should be improved or modified in future. We also introduce important features about key management of digital signature; the approach of it could be used to improve the security of information transfer between community parties.

Finally, we summarize the main approaches in our study. Since all kinds of software, applications are available with widely developing and downloading from the Internet, the interactions between all kinds of users becomes universal. The reputation of software is one of the most important features in evaluating software and it affects people’s daily use and experience about different kinds of software. Through our study towards this key point, we have proposed and analyzed methods to rate software and keep the objectivity of their reputation calculation results.


