Global Journals LaTeX JournalKaleidoscope™
Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.*

Assessing the Quality of a Software Service at the Time of Project Development by Identifying its Reputation

Dr. Panchamukesh Ch[1] and Venkateswarlu B[2]

1

Received: 24 August 2011 Accepted: 19 September 2011 Published: 29 September 2011

Abstract

At the time of integration of the software while developing a project the reputation and the quality of execution is tough to identify and which is very risky. As the software industry is introduced with a new type of service delivery model known as SaaS(Software as a service),the problem has increased a lot . Existing system be inclined to rely on rating from customer to experiences of past service which may create major issues in terms of subjectivity and rating unfairness. Few previous works have been considered quality and reputation for selection of services bur none have done service rating process through automation. We proposed an automated quality and reputation framework for rating and selecting a service. In this paper the management of risk has been formulated in context of development of the project using third party software service components and credibility is calculated by a measured reputation system.

Index terms— Reputation, Service Vendor, Automation, SaaS, Service rating.

1 INTRODUCTION

Author ? : M.Tech, Department of Information Technology. That's the reason why software industry has started moving toward a new kind of software delivery model called SaaS(Software as a Service) and which made the things easy to install, maintenance-free, and money-spinning. In Software as a Service (SaaS) software delivery model the software is delivered ondemand and priced on-use, which made it to be widespread implementation of fast Internet access, combined with the widespread acceptance of SOA based solutions. SaaS has gained popularity by reducing the cost of tenure and alleviating the burden of software installation and maintenance. SaaS contributions has expanded dramatically as some of the enterprises have started to outsource their software infrastructure and development projects to SaaS vendors, and the competition has been increased even among vendors of traditional on premises software as in fig 1.

In the world of Software development using service delivery by SaaS model the quality of the software and software provider's credibility is tough and risky. So, the integration of external software in project development is challenging.In this paper risk management has been addressed in context of project . he application of a quantifiable, systematic, disciplined approach to the development, process, and maintenance of software can be stated as software engineering. As software industry has huge competition it has shaped a strong motivation for developing solutions to support more responsive and more competitive businesses. Even with long-standing success of COTS (commercial off-the-shelf) software as a time-effective alternative to custom "in-house" developed solutions is still being compromised by the implicated cost of ownership, installation and maintenance time, and effort. T E-mails : mukesh_1229@yahoo.com, iambondu@gmail.com basis of fair and objective feedbacks. Most of the works that addressed until now are on evaluating the fairness of existing Feedbacks. Work in this paper focuses instead on the process of generating objective and fair feedbacks. Feedback can be individual since it is based on consumers' "personal" expectations and opinions. Consumers may have an obstructed view of a service reputation systems are prone to attacks by malicious consumers who may give false ratings and subvert

service reputation. Consumers may have little incentive to leave a feedback. In this perspective a framework an automated quality and Reputation based framework for service rating and selection has been proposed.

The main objectives of this paper are: a) In order for a reputation mechanism to be fair and objective, it is essential to compute reputation on the basis of fair and objective feedbacks. b) The simulation results have demonstrated that the devised system has successfully met our primary objectives and can be an important component in a risk management strategy for software development with SaaS. c) A computational model is provided to objectively evaluate the delivered service based on the actual measurement of the conformance of the execution quality to the contracted SLA. A novel algorithm is also devised to automate the rating process based on the expectancy-disconfirmation theory from market science.

II.

2 RELATED WORK

What is the main correlation stuck between "reputation" and "trust"? The major difference between reputation and trust can be illustrated by the following statements: (a) "Because of your good reputation I trust you" (b) "I trust you despite your bad reputation." Here the reputation is a collective measure of trustworthiness and is measured based on the referrals or ratings from other members in a community. According to A.josang and R.ismail, reputation is believed about a person's or thing's character or standing. Hence, trust for an individual is measured from the personal reputation and In a centralized reputation management system, the synthetic rating of QoS of web services is aggregated by each rating in the community. To avoid the inapt evaluation by dishonest consumers, it need identify the reputable and disreputable members with their historical comments. Our idea is that consumer reputation is decided by the historical quality of comment, that is, more positive comments gain higher reputation, versa. In other words, lower reputations will worse his/her performance rating on QoS evaluation of web services. When consumers jointing the voting activity can raise their reputation by positive comments and avoid the negative comments. In this work, we proposed a centralized reputation measure for quantifying consumer reputation to properly select the service alternatives, as illustrated in Fig 3.

3 III. SYSTEM ANALYSIS & DESCRIPTON

For selection of the service many previous works have measured the reputation and quality of the software, but the measurement has been done using some manual tools but none have considered the service rating process in the form of automation. WE introduce a framework for selecting and rating software to provide software service. The important point of the framework which is proposed is to automate both the rating and selection software services which is potentially increasing the objectivity of the service quality reports and concentrating on time-consumption and which finally reduces the risk associated utilization of external software services in development projects. While determining a service's suitability to a particular user's preferences in terms of quality and cost the service selection algorithm acts as a user-centric and reputation-aware service recommender. In order for a reputation mechanism to be fair and objective, it is essential to compute reputation on the basis of fair and objective feedbacks. Our work focuses instead on the process of generating objective and fair feedbacks, while most of the works that addressed this latter issue are on evaluating the fairness of existing feedbacks.

4 Global Journal of Computer

Here concentrated the calculation of the reputation on works in the area of Service Level Agreement (SLA) monitoring where a computational model is provided to neutrally assess the delivered service based on the actual measurement of the execution quality to the contracted SLA.

In this paper we proposed a framework which has four major modules like Consumer, SLA (Service Level Agreement), Service Providers and Reputational System. Consumer can start the selection based on the trustworthiness features.

Consumer selection information will be stored inside database like reputation table. SLA maintains some of the requirements about that particular service. These requirements can be coinside with SLA requirements and for those services only the service certificate will be approved and that Certificate can be used as Trustworthiness certificate. The services which are provided by SLA can also be present in the service providers itself. User can be satisfied with certified services or trustworthy services. All the user behaviors features can be located inside the trustworthy services. To start the selection at the consumer side we should place the all the features inside that particular service. Reputation can be defined based on the frequent item selection procedure to define the utility measurement identification. Based on utility measure the feedback about that particular service will be defined. The proposed reputational framework is as shown in fig 4 **??** And the functional requirements of the proposed frame work will be as Enter Consumer Details, Update Consumer Required Services, and Enter Service Provider Details, Service updated to SLA, Retrieve Services, Select Service, Utility Measure of Service, Rating Function, Retrieve Feedback, Consumer Preference Updated, Select service and Calculate Score. An empirical study of the risk factors related to the development using external software (COTS-like) components along with associated risk reduction activities has been reported in. It showed that risk reduction at software selection time is negatively correlated with occurrences of most project development-related risks. In fact, selection must be driven by quality constraints, with selection time evaluation of component quality and choice of appropriate

service providers all essential to successful integration. However, in practice, the evaluation of service quality cannot be performed until the service is acquired. Consequently, quality evaluation is typically limited to the evaluation of quality offers by comparing the quality level that providers promise to the quality requirements. Compliance cannot be guaranteed at selection time, so it is essential to choose a provider that is trusted to respect its commitments.

5 IV. SYSTEM DESIGN & IMPLEMENTATION a) Designing of the Framework

In the system design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the static relations between those objects. With detailed modeling, the classes of the conceptual design are often split in a number of subclasses.

In order to further describe the behavior of systems, these class diagrams can be complemented by state diagram or UML state machine. Where in our framework we have four classes Service Provider, SLA, Consumer and reputational System as in Fig 5 **??**Here Service provider will check for the service name, cost of service, utility of service and value of time. In SLA class it will monitor the service and measures the utility and produces the rating function and identifies the feedback. In Consumer class consumer will select the category, finds utility, cost and selects the service. In Reputation System it identifies the user preferences then select the service and maintains the time. A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. Use Case diagrams are formally included in two modeling languages defined by the OMG: the Unified Modeling Language (UML) and the Systems Modeling Language (SysML).Major two components for a use case diagram are as follows:

? Use cases A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse. Where in our Framework we have set of use-cases like Service Name, Category, Cost of Service, Utility, Value of Time, Monitoring the services, Service Rating, User Preferences, Reputation, Feedback as in Fig 6 **??** ? Actors An actor is a person, organization, or external system that plays a role in one or more interactions with the system. Where in our framework we have Service Provider and Consumer as actors. Step 3: Update the Consumer Required Services.

Step 4: Service will be updated into SLA.

Step 5: Consumer will retrieve the services.

Step 6: Consumer choose to select the service.

Step 7: Measurement of Utility for the service.

Step 8: Calculating the Rating Function.

Step9: Retrieving the Feedback for the Service selected.

Step 10: Consumer Preference will be updated.

Step 11: Calculate the Score depending the selected service.

Step 12: Select the service depending on the score achieved. Step 13: Stop.

V.

6 RESULTS

The following are the screen shots of the system.

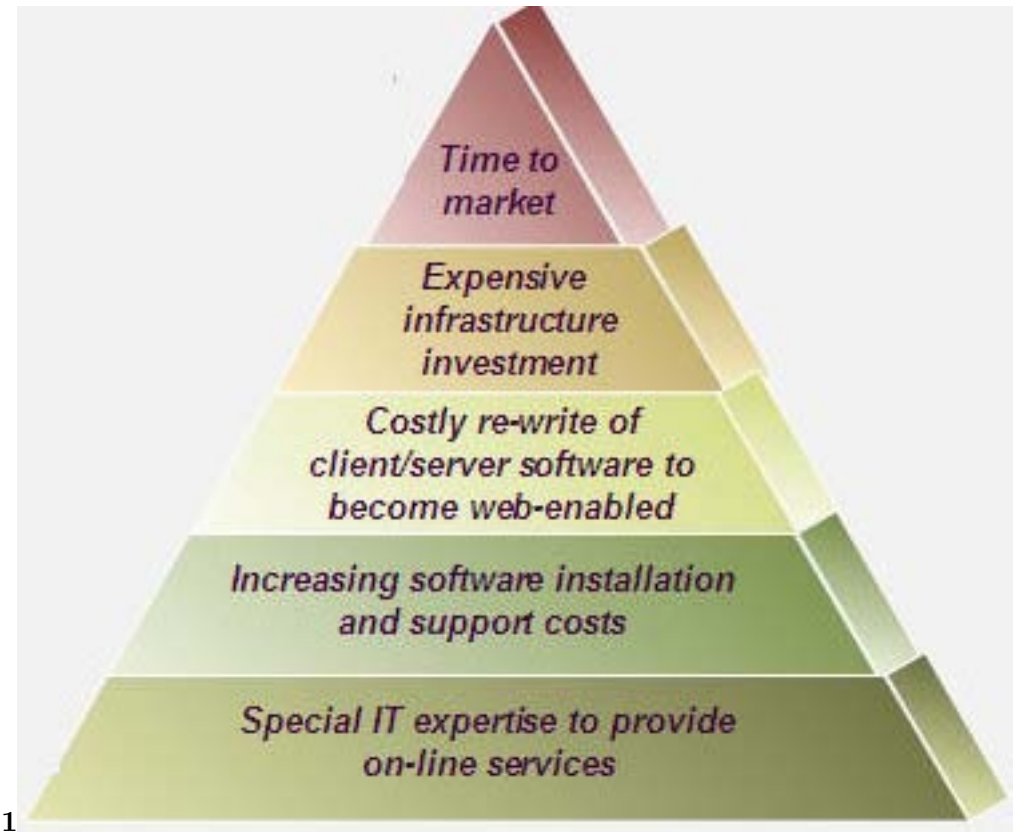
.

7 CONCLUSION

In this paper we addressed the risk of incorporating a third party software for development of a project. To overcome the risk factor, proposed an outstanding framework Identifying the Reputation and Assessing the Quality of a Software Service at the Time of Project Development. We highlighted the framework by adding enhanced features like consumer, SLA, Service Provider and Reputation System which made as added additional advantage in rating and selecting the software to be used for integration. The proposed framework have accomplished in confining the service behaviors and translating them into probable customers choice.

[1] [2]

Figure 1: ?



1

Figure 2: Fig 1 :

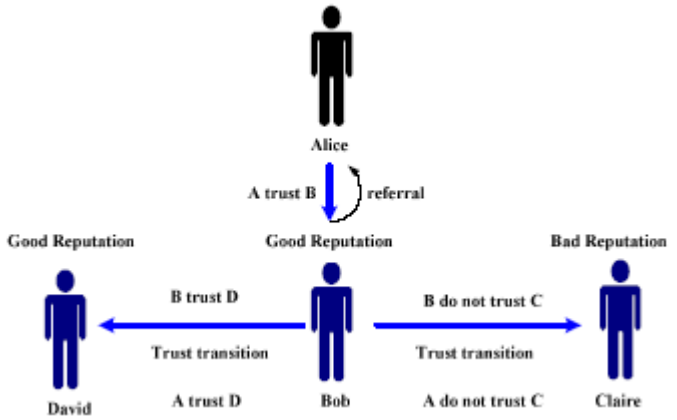


Figure 3:

2

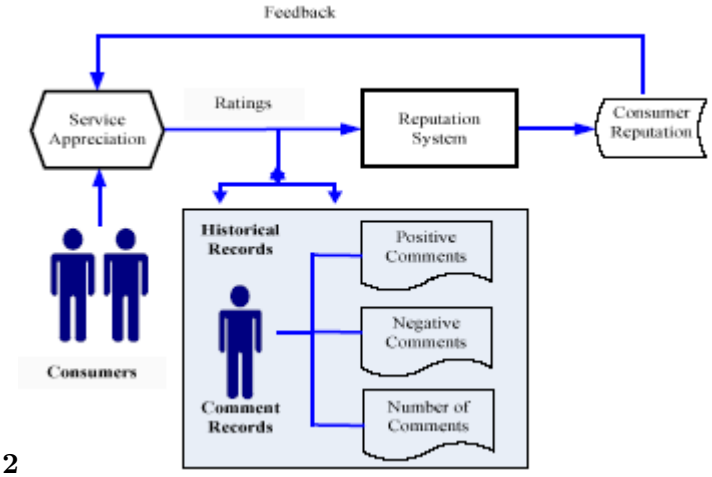
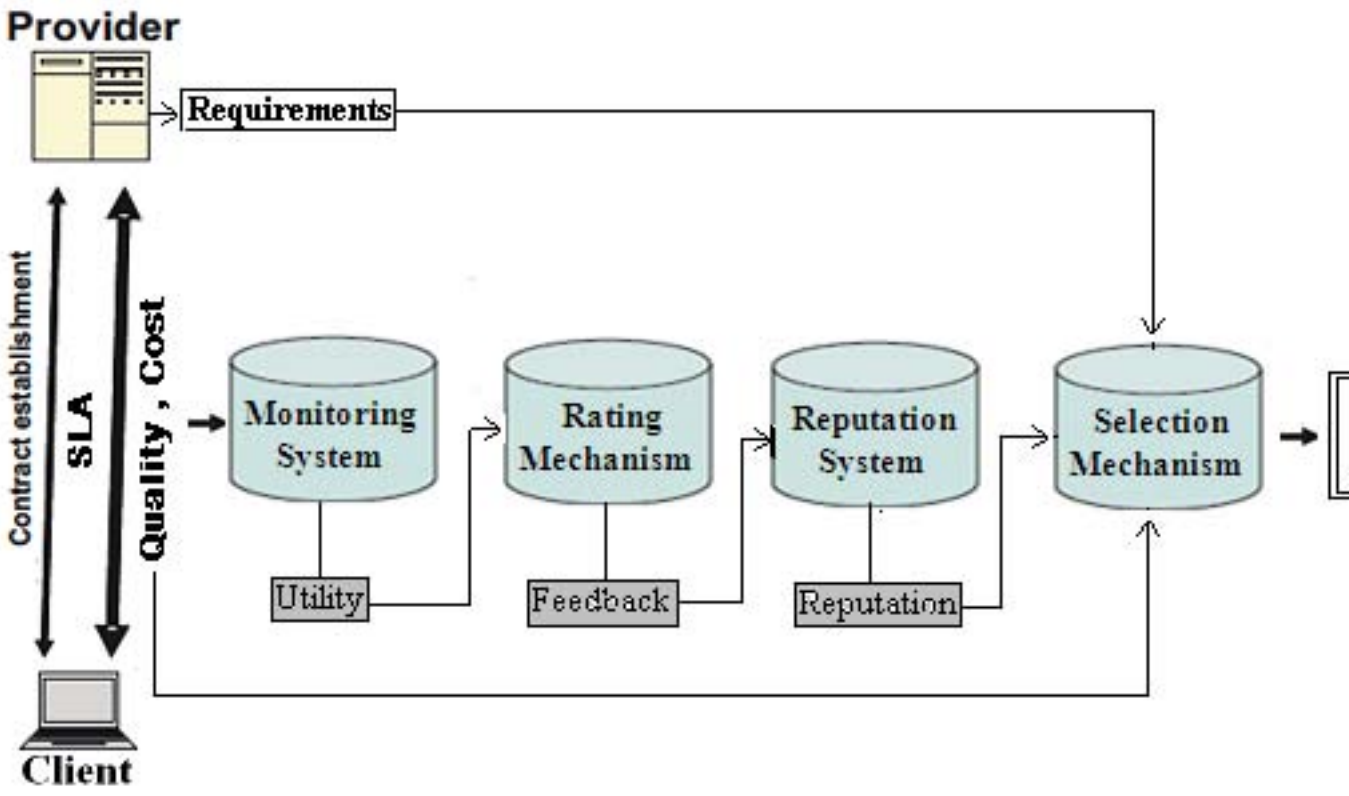


Figure 4: Fig 2 :



3

Figure 5: Fig 3 :

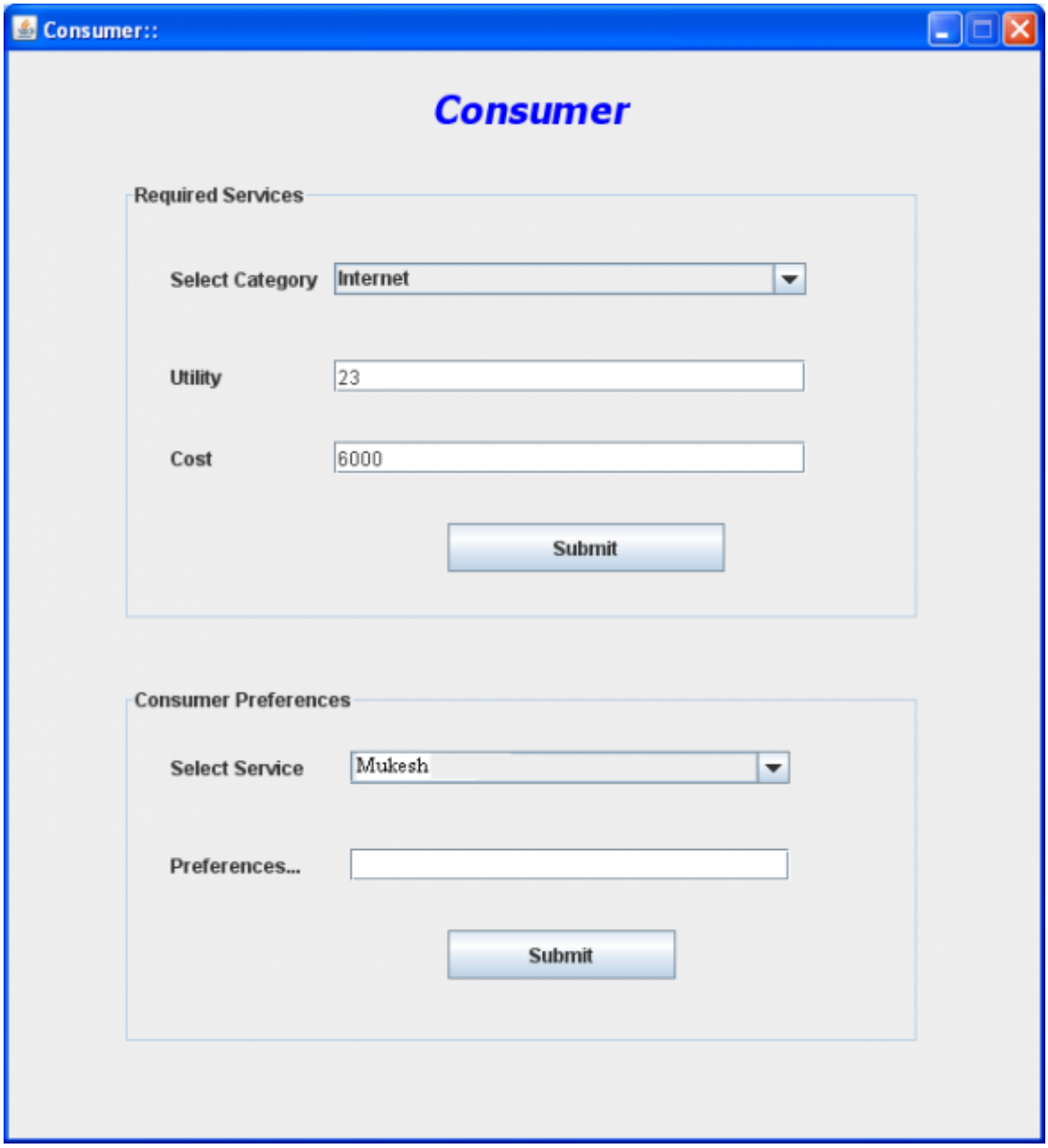
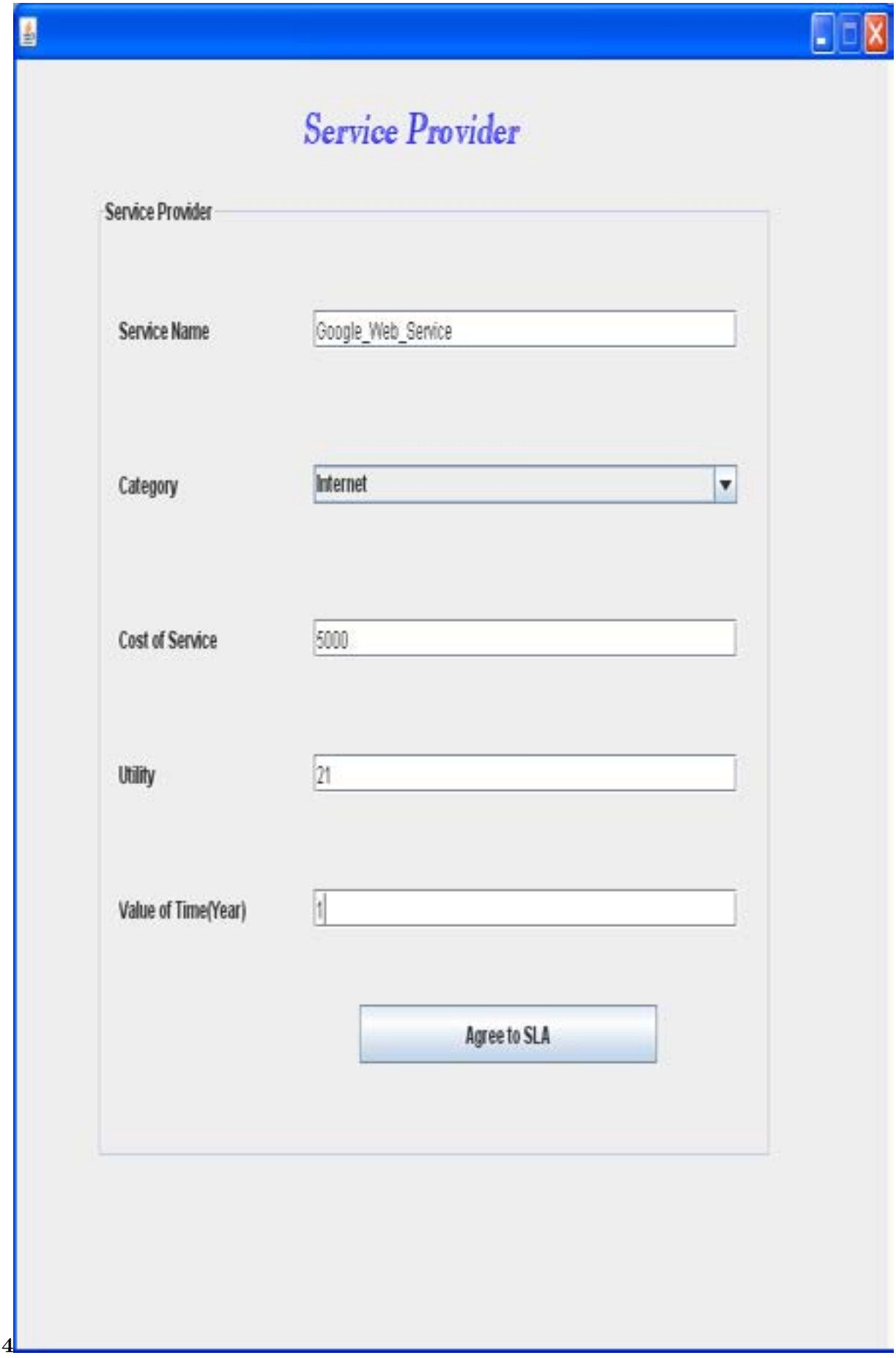
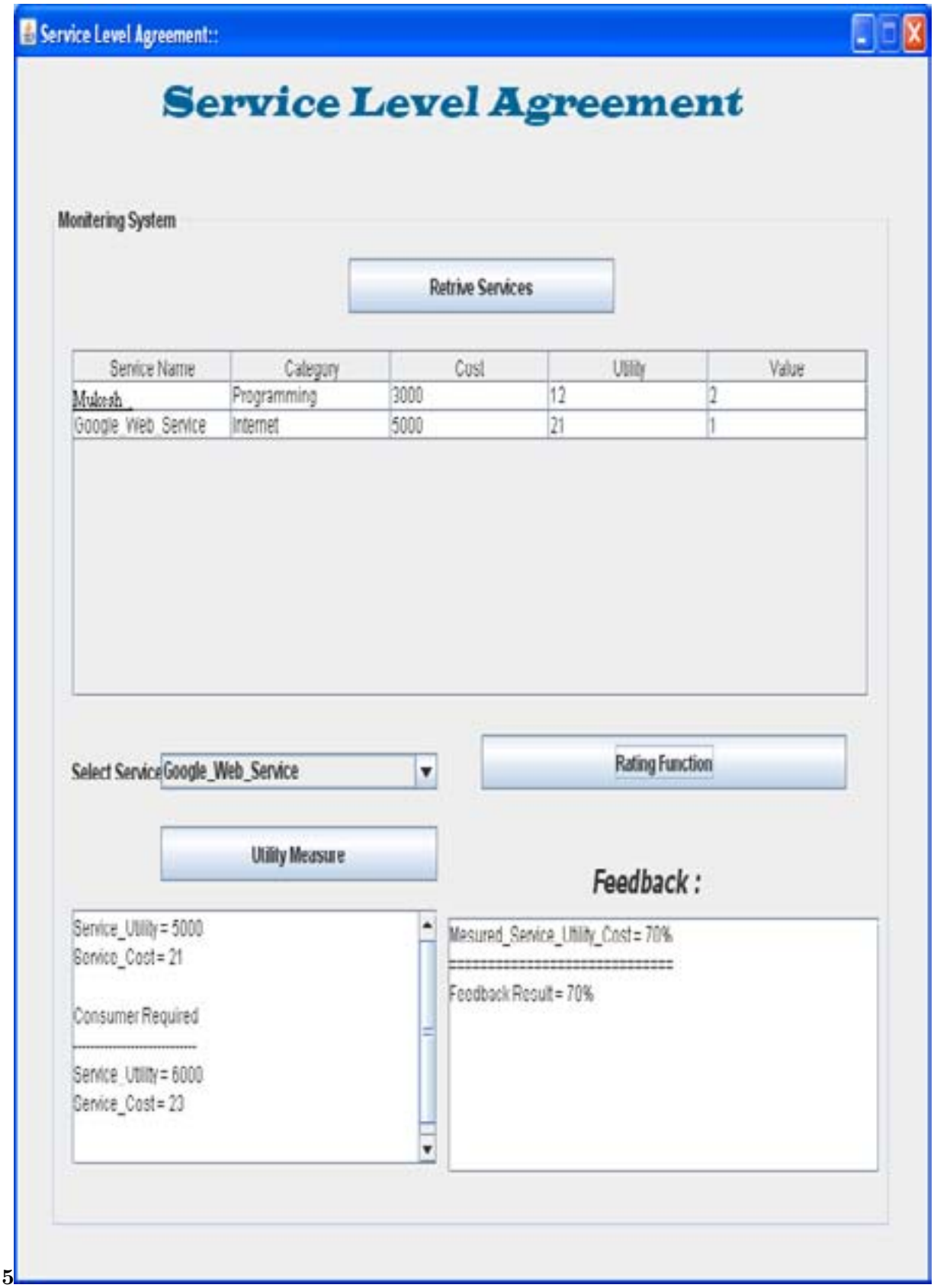

Consumer::
Consumer
Required Services
Select Category
Internet
Utility
23
Cost
6000
Submit
Consumer Preferences
Select Service
Mukesh
Preferences...
Submit

Figure 6:



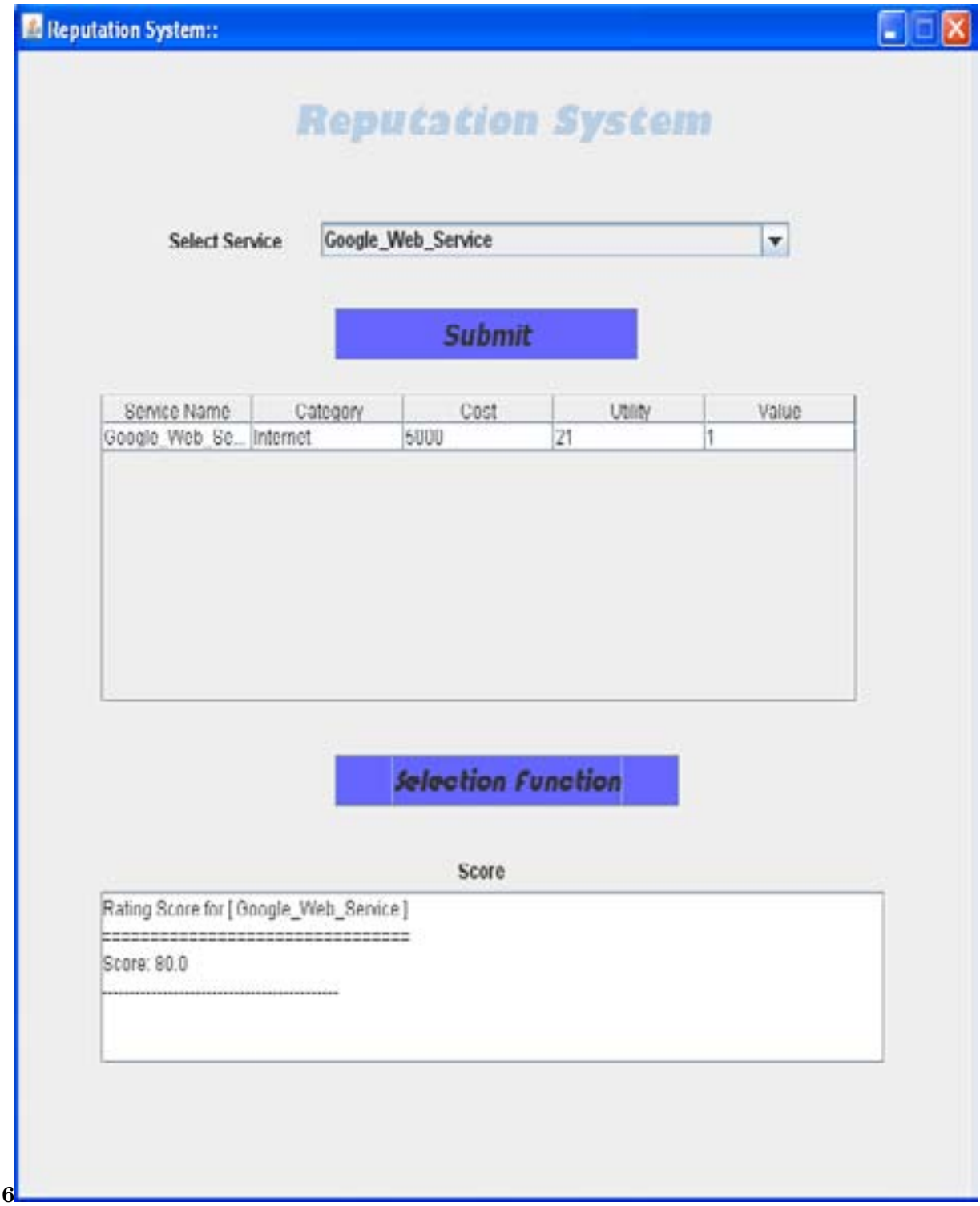
4

Figure 7: Fig 4 :



5

Figure 8: Fig 5 :



6

Figure 9: Fig 6 :

153 [Eng (2008)] , Software Eng . Mar./Apr. 2008. 34 p. .

154 [Anselmi et al. ()] 'A QoS-Based Selection Approach of Autonomic Grid Services'. J Anselmi , D Ardagna , P
155 Cremonesi . *Proc. Workshop Service-Oriented Computing Performance: Aspects, Issues, and Approaches*,
156 (Workshop Service-Oriented Computing Performance: Aspects, Issues, and Approaches) 2007.

157 [Li et al.] 'A State-of-the Practice Survey of Risk Management in Development with Off-the-Shelf Software
158 Components'. J Li , R Conradi , O P Slyngstad , M Torchiano , M Morisio , C Bunse . *IEEE Trans*

159 [Jøsang et al. ()] 'A survey of trust and reputation systems for online service provision'. A Jøsang , R Ismail , C
160 Boyd . *Decision Support Systems* 2007. 43 (2) p. .

161 [Zeng et al. (2004)] 'QoS-Aware Middleware for Web Services Composition'. L Zeng , B Benatallah , A H Ngu ,
162 M Dumas , J Kalagnanam , H Chang . *IEEE Trans. Software Eng* May 2004. 30 (5) p. .

163 [Vu et al. ()] 'QoS-Based Service Selection and Ranking with Trust and Reputation Management'. L.-H Vu ,
164 M Hauswirth , K Aberer . *Proc. 13th Conf. Cooperative Information Systems*, (13th Conf. Cooperative
165 Information Systems) 2005.

166 [Saas ()] T Saas . http://trustsaas.com/ *Trust Saas: Putting the Trust in Software as a Service (SaaS)*,
167 2008.

168 [Belgonet ()] 'Service Level Agreement'. Belgonet . http://www.easyservermanagement.com/sla.php
169 *EZSM Service Level Agreement* 2008. 2008.

170 [Skene et al. (2010)] 'Service-Level Agreements for Electronic Services'. J Skene , F Raimondi , W Emmerich .
171 10.1109/TSE.2009.55. http://doi.ieeecomputersociety.org/10.1109/TSE.2009.55 *IEEE Trans.*
172 *Software Eng* Mar./Apr. 2010. 36 (2) p. .

173 [Skene et al. (2010)] 'Service-Level Agreementsfor Electronic Services'. J Skene , F Raimondi , W Emmerich .
174 10.1109/TSE.2009.55. http://doi.ieeecomputersociety.org/10.1109/TSE.2009.55 *IEEE Trans.*
175 *Software Eng* Mar./Apr. 2010. 36 (2) p. .

176 [Skene et al. ()] 'The Monitorability of Service-Level Agreements for Application-Service Provision'. J Skene ,
177 A Skene , J Crampton , W Emmerich . *Proc. Sixth Int'l Workshop Software and Performanc*, (Sixth Int'l
178 Workshop Software and Performanc) 2007. p. .

179 [Douceur ()] 'The Sybil Attack'. J R Douceur . *Proc. First Int'l Workshop Peer-to-Peer Systems*, (First Int'l
180 Workshop Peer-to-Peer Systems) 2002.

181 [Keller and Ludwig ()] 'The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web
182 Services'. A Keller , H Ludwig . *J. Network and System Management* 2003. 11 (1) p. .