Volume 2, Issue 8, August 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Analysing & Identifying SAAS for Development of A Project by Calculating its Reputation

Mrs. *Neelima Sadineni,**
* MTech(Information technology) Student ,
GOKUL Engineering College, Bobilli
AndhraPradesh, INDIA.

Mr. W. Anil **
** Asst.Professor , Dept if CSE,
GOKUL Engineering College, Bobilli
Andhra Pradesh, INDIA

Abstract— Assessing the quality of external software before integrating it in to the project development is very challenging now a day. As IT industry is moving towards newly evolving tool named SaaS(Software as a Service) , the risk of integrating the external software to the project development has been increased . Presently integration of external software is going on, but they use the traditional way of collecting the feedbacks to identify whether to use that external software into the project or not, which may produce an unfair results at the end of project deployment. So in this perspective we are going to propose an automated framework to rate and select a service by identifying quality and reputation .And we mainly focused on addressing the risk in proposing external software by using quality and reputation of it.

***Keywords**— Software as a Service, Quality, Project development, Automation, risks*.

1. INTRODUCTION

Software Engineering can be stated in Industrial terms as the application of a quantifiable, systematic, disciplined approach to the development, process, and maintenance of software[4]. As software industry has huge competition it has shaped a strong motivation for developing solutions to support more responsive and more competitive businesses [1][8]. 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[6][9][11]. 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 on-demand and priced on-use, which made it to be widespread implementation of fast Internet access, combined with the widespread acceptance of SOA based solutions[5].

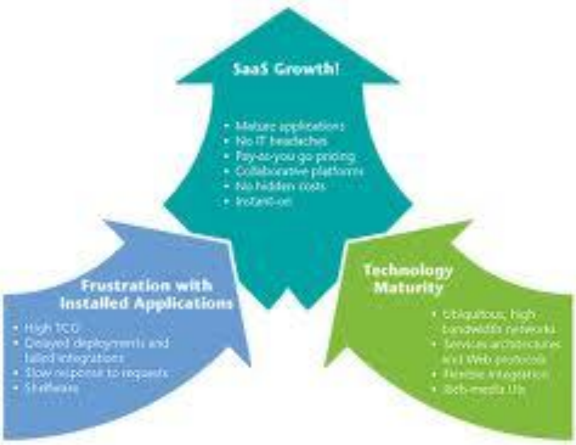

SaaS Growth!
• Mature applications
• No IT headaches
• Pay-as-you go pricing
• Collaborative platforms
• No hidden costs
• Instant-on
Frustration with Installed Applications
• High TCO
• Delayed deployments and failed integrations
• Slow response to requests
• Shelfware
Technology Maturity
• Ubiquitous, high bandwidth networks
• Services architectures and Web protocols
• Flexible integration
• Rich-media UIs

Fig 1: The Growth of SaaS

SaaS has gained popularity by reducing the cost of tenure and alleviating the burden of software installation and maintenance[2]. 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[1][3] 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[2]. So, the integration of external software in project development is challenging. In this paper risk management has been addressed in context of project development using external software service components. Reputation must be computed on the basis of fair and objective feedbacks. Most of the works that addressed until now are on evaluating the fairness of existing Feedbacks[7]. 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[13]. 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 as in Fig.2.

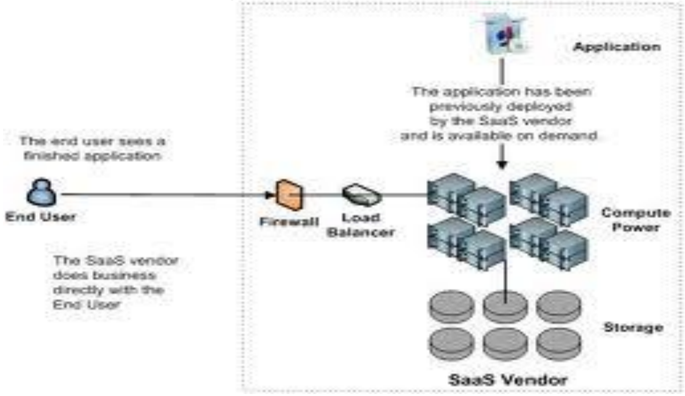

Application
The application has been previously deployed by the SaaS vendor and is available on demand
The end user sees a finished application
End User
Firewall
Load Balancer
Compute Power
The SaaS vendor does business directly with the End User
Storage
SaaS Vendor

Fig.2: The Role of Software Vendors in Implementing SaaS.

The challenges 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.

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[14][15][16]. 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 combination of received referrals, as in the Fig 3.

In a centralized reputation management system, the synthetic rating of QoS of web services is aggregated by each rating in the community[4]. To avoid the inapt evaluation by dishonest consumers, it need identify the reputable and disreputable members with their historical comments.

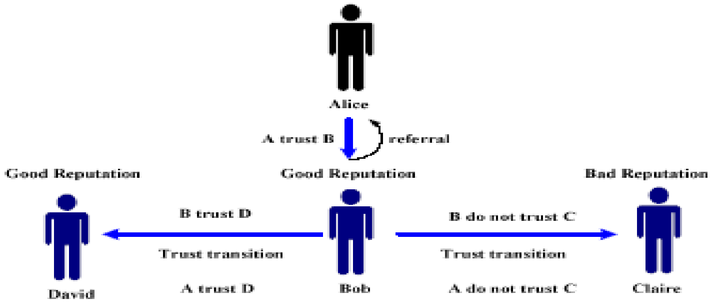


Fig 3: A Transitive model for consumer reputation

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[11][10]. 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 4.

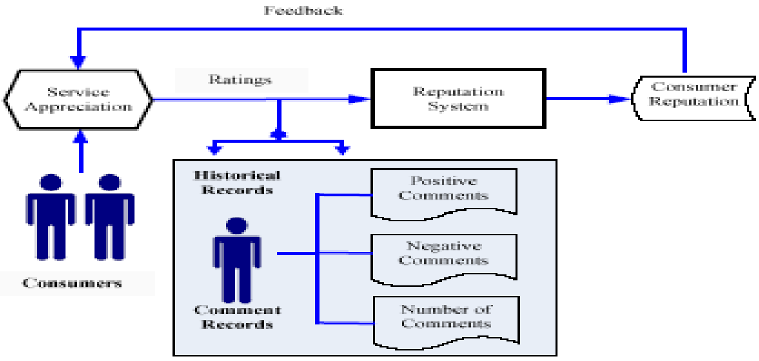


Fig 4: Consumer Reputation Measure

3. ARCHITECTURE OF THE PROPOSED

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.

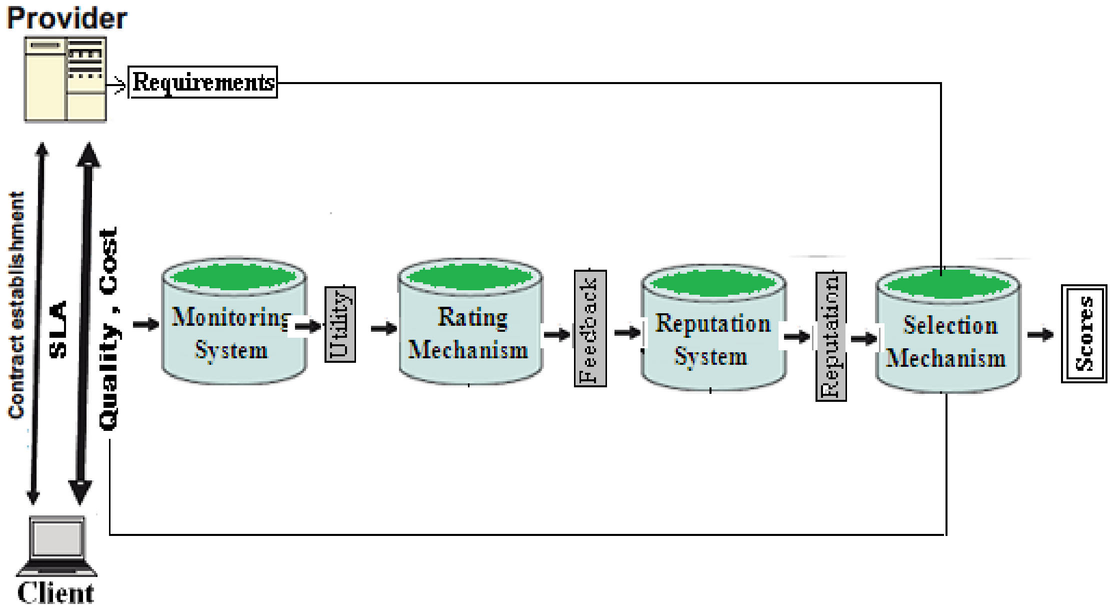


Fig 5: System Architecture of the framework

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. 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 co-inside 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

4. DESIGN & IMPLEMENTATION OF THE SYSTEM

a) Designing of the Framework

In the context of software, design is problem solving process whose objective is to find and describe a way to find and describe the way to implement the functional requirements while respecting the constraints imposed by the non functional requirements and by adhering to general principles of good quality. The goal of the design process is to produce a model or representation of a system which can be used later to build that system and use this model to build the overall system.

Design is concerned with identifying software components specifying relationships among components. Specifying software structure and providing blue print for the document phase. Modularity is one of the desirable properties of large systems. It implies that the system is divided into several parts. In such a manner, the interaction between parts is minimal clearly specified. Design will explain software components in detail. This will help the implementation of the system. Moreover, this will guide the further changes in the system to satisfy the future requirements.

Class Diagram contains the following elements:

- A class which represents entities with common characteristics or features attributes operations and associations.
- Association, which represent relationship between two or more classes where relationships have common characteristics or features like attributes and operations as in Fig.6.

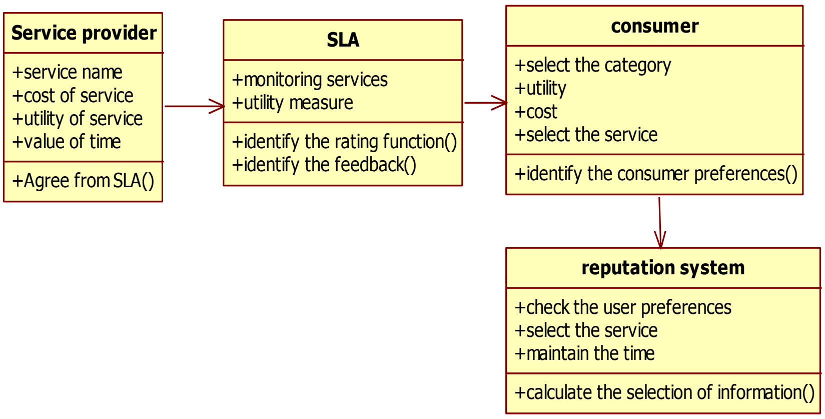


Fig 6: Inter-operational Class diagram for framework

Sequence Diagrams represents the interactions between classes to achieve a result such as a usecase. The sequence diagram lists objects horizontally and time vertically, and models these messages over time. In this paper the designing of the sequence has been done on both service provider and the consumer and they were described as in the Fig.7 and Fig .8

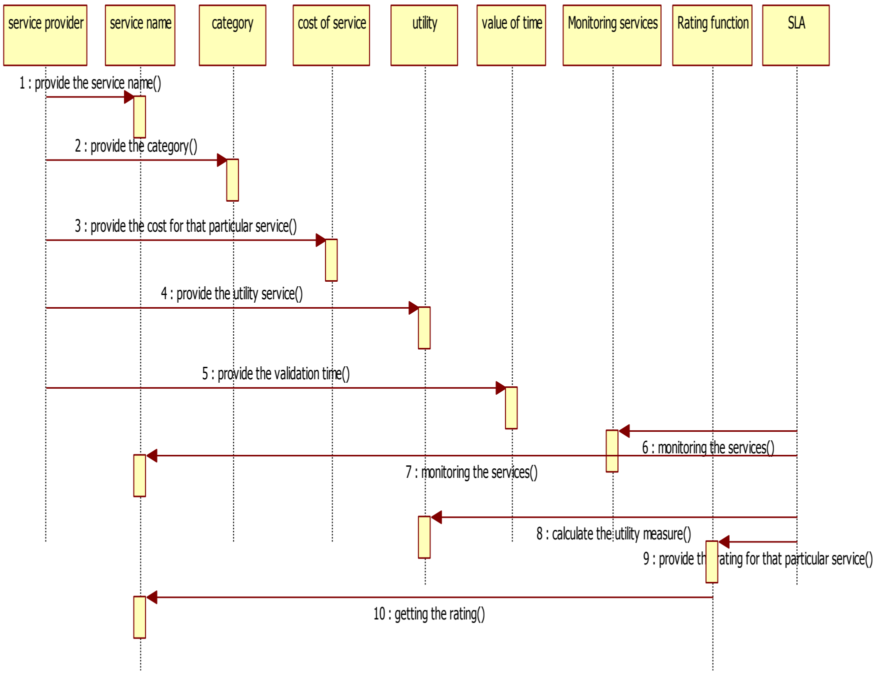

service provider
service name
category
cost of service
utility
value of time
Monitoring services
Rating function
SLA
1 : provide the service name()
2 : provide the category()
3 : provide the cost for that particular service()
4 : provide the utility service()
5 : provide the validation time()
6 : monitoring the services()
7 : monitoring the services()
8 : calculate the utility measure()
9 : provide the rating for that particular service()
10 : getting the rating()

Fig 7: Inter-operational Sequence diagram for the Service Provider

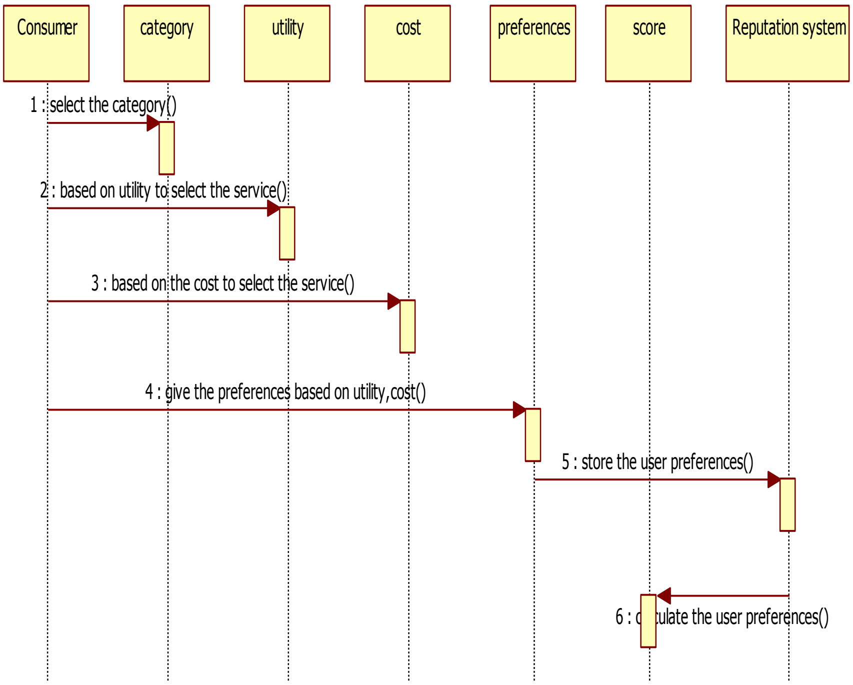

Consumer
category
utility
cost
preferences
score
Reputation system
1 : select the category()
2 : based on utility to select the service()
3 : based on the cost to select the service()
4 : give the preferences based on utility,cost()
5 : store the user preferences()
6 : calculate the user preferences()

Fig 8: Inter-operational Sequence diagram for the Consumer

b) Flowchart Representation of Framework

The Flowchart representation of the framework is detailed below:

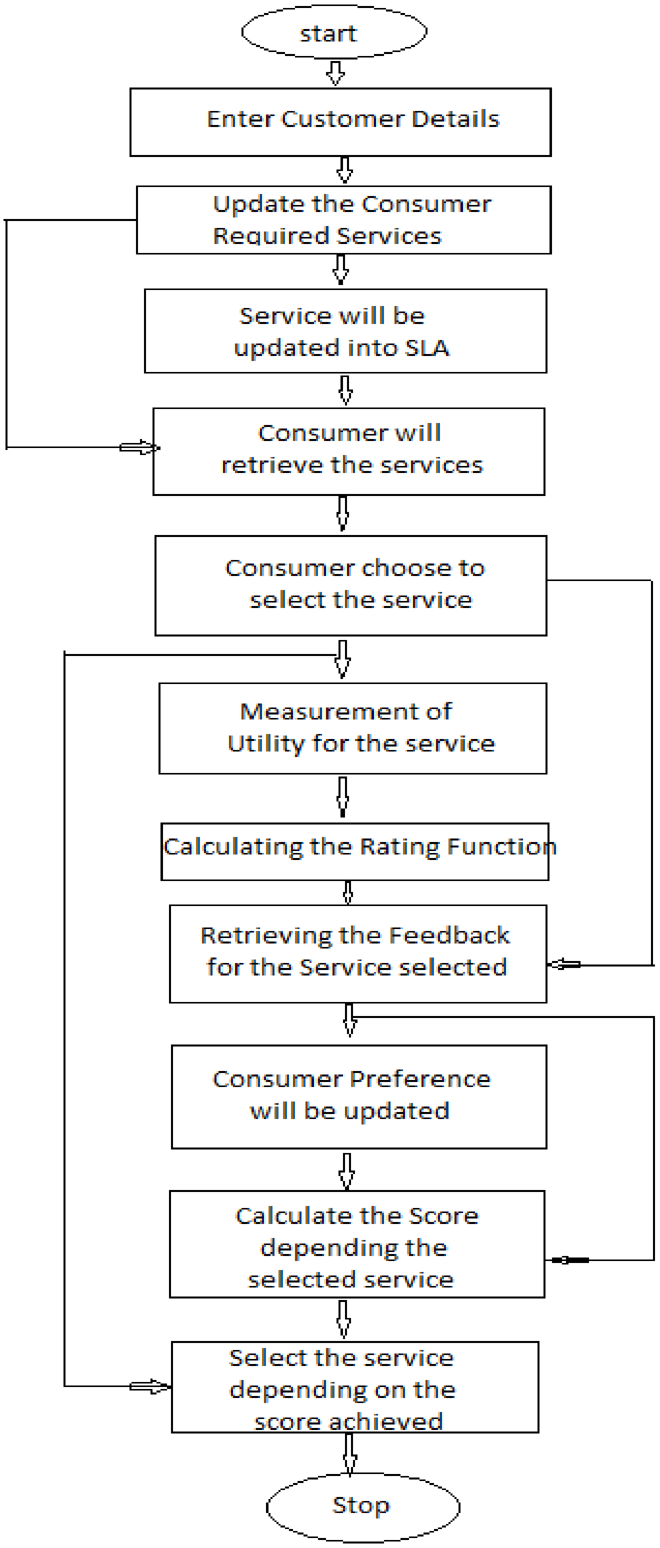
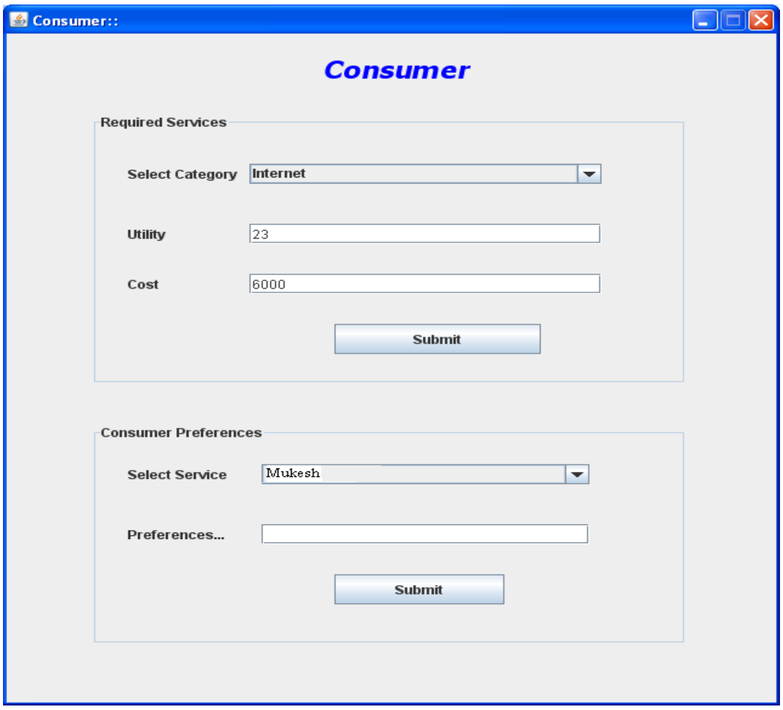

start
Enter Customer Details
Update the Consumer Required Services
Service will be updated into SLA
Consumer will retrieve the services
Consumer choose to select the service
Measurement of Utility for the service
Calculating the Rating Function
Retrieving the Feedback for the Service selected
Consumer Preference will be updated
Calculate the Score depending the selected service
Select the service depending on the score achieved
Stop

Fig.9: Flowchart Representation of the System.

5. RESULTS

Obtained results of the system are displayed as screen shots in the following section.



Consumer::

Consumer

Required Services

Select Category: Internet

Utility: 23

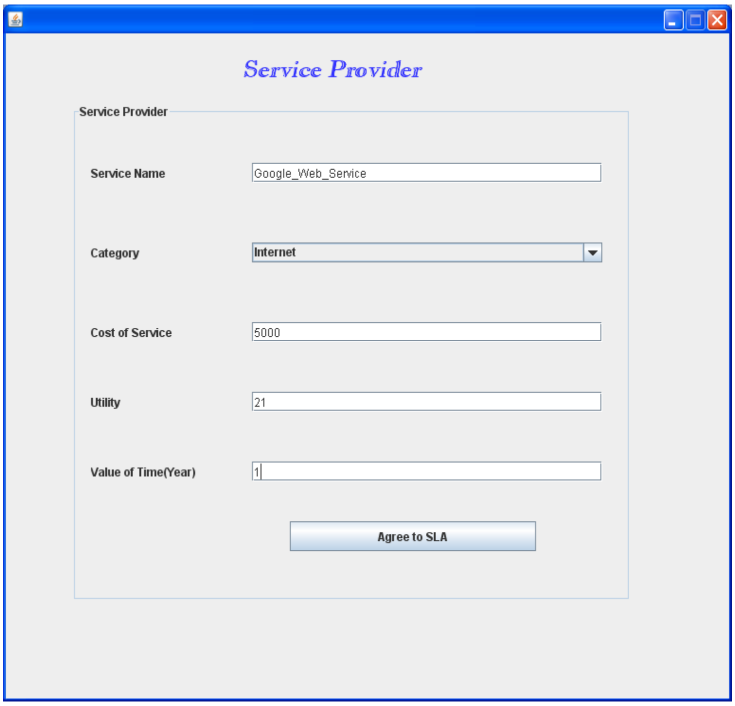
Cost: 6000

Consumer Preferences

Select Service: Mukesh

Preferences...

Fig.10: Selection of required service by consumer



Service Provider

Service Provider

Service Name: Google_Web_Service

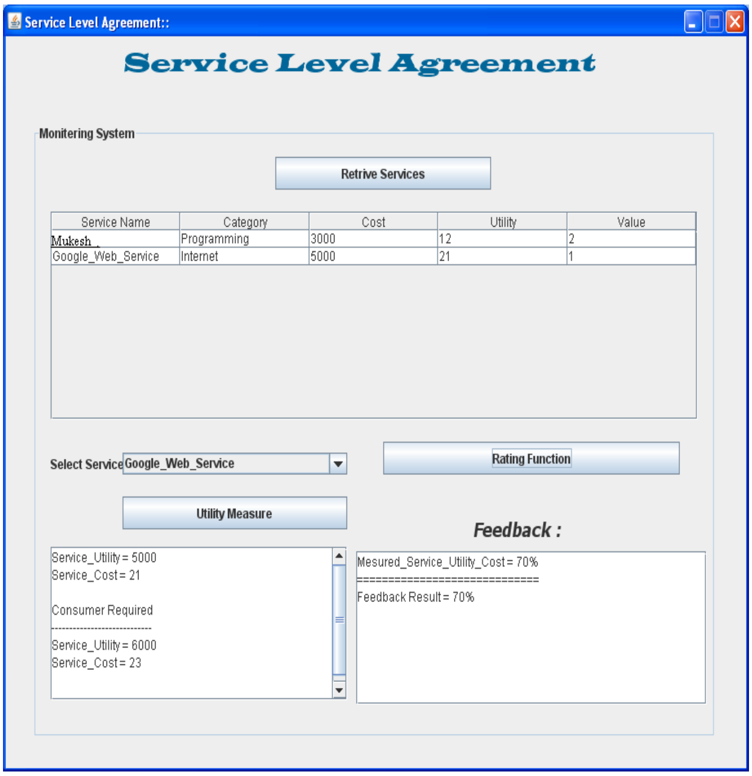
Category: Internet

Cost of Service: 5000

Utility: 21

Value of Time(Year): 1

Fig.11: Updating the type of service provided by service provider



Service Name	Category	Cost	Utility	Value
Mukesh	Programming	3000	12	2
Google_Web_Service	Internet	5000	21	1

Select Service Google_Web_Service

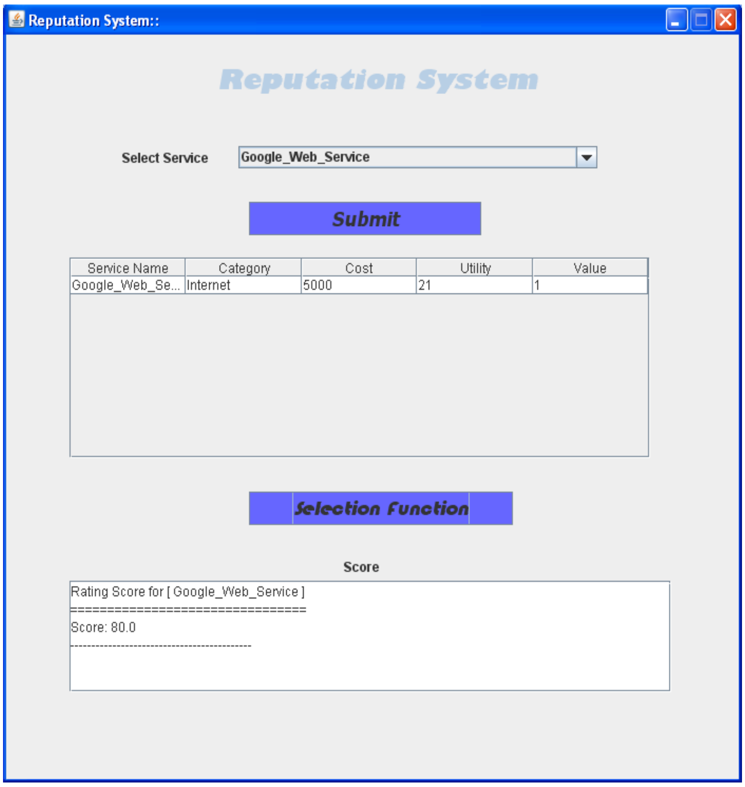
```
Service_Utility= 5000
Service_Cost= 21

Consumer Required
---------------------------
Service_Utility= 6000
Service_Cost= 23
```

Feedback :

```
Mesured_Service_Utility_Cost= 70%
============================
Feedback Result= 70%
```

Fig.12: The SLA between Consumer and Service Provider



Select Service Google_Web_Service

Service Name	Category	Cost	Utility	Value
Google_Web_Se...	Internet	5000	21	1

Score

```
Rating Score for [ Google_Web_Service ]
================================
Score: 80.0
-----------------------------------------------
```

Fig.13: Calculating the Score of the service using Reputation System

CONCLUSION

In this paper we addressed the risk to incorporate third party software for project development. As we identified the integration of it is very risky we proposed an outstanding automated framework to rate and select a service by identifying quality and reputation. 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.

REFERENCES

1. Panchamukesh Chandaka , Venkateswarlu B , "Assessing the Quality of a Software Service at the Time of Project Development by Identifying its Reputation ", Published at Global Journal of Computer Science and Technology Volume 11 Issue 18 Version 1.0 October 2011.
2. Panchamukesh Chandaka , Shankar K ,"A reputational framework for rating and selecting software service at the time of integration" ,Panchamukesh Ch* et al. / (IJAEST) INTERNATIONAL JOURNAL OF ADVANCED ENGINEERING SCIENCES AND TECHNOLOGIES Vol No. 8, Issue No. 2, 214 - 218
3. Panchamukesh Chandaka , Venkateswarlu B , "Assessing the Quality of a Software Service at the Time of Project Development by Identifying its Reputation " Published at IJCEM International Journal of Computational Engineering & Management, Vol. 14, October 2011 ISSN (Online): 2230-7893
4. A. Jøsang, R. Ismail, C. Boyd, “A survey of trust and reputation systems for online service provision”, Decision Support Systems, Vol.43, No.2, 2007, pp.618-644.
5. J. Li, R. Conradi, O.P. Slyngstad, M. Torchiano, M. Morisio, and C.Bunse, “A State-of-the Practice Survey of Risk Management in Development with Off-the-Shelf Software Components,” IEEE Trans.
6. T. SaaS, “Trust Saas: Putting the Trust in Software as a Service (SaaS),” http://trustsaas.com/, 2008.
7. J. Skene, F. Raimondi, and W. Emmerich, “Service-Level Agreements for Electronic Services,” IEEE Trans. Software Eng., vol. 36,no. 2, pp. 288-304, Mar./Apr. 2010, http://doi.ieeecomputersociety.org/ 10.1109/TSE.2009.55.
8. A.Keller and H. Ludwig, “The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services,”J. Network and System Management, vol. 11, no. 1, pp. 57-81, 2003.
9. Software Eng., vol. 34, no. 2, pp. 271-286, Mar./Apr. 2008.
10. J. Anselmi, D. Ardagna, and P. Cremonesi, “A QoS-Based Selection Approach of Autonomic Grid Services,” Proc. Workshop Service-Oriented Computing Performance: Aspects, Issues, and Approaches, 2007.
11. L. Zeng, B. Benatallah, A.H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, “QoS-Aware Middleware for Web Services Composition,” IEEE Trans. Software Eng., vol. 30, no. 5, pp. 311-327, May 2004.
12. L.-H. Vu, M. Hauswirth, and K. Aberer, “QoS-Based Service Selection and Ranking with Trust and Reputation Management,” Proc. 13th Conf. Cooperative Information Systems, 2005.
13. J. Skene, F. Raimondi, and W. Emmerich, “Service-Level Agreementsfor Electronic Services,” IEEE Trans. Software Eng., vol. 36,no. 2, pp. 288-304, Mar./Apr. 2010, http://doi.ieeecomputersociety .org/10.1109/TSE.2009.55.
14. J.R. Douceur, “The Sybil Attack,” Proc. First Int’l Workshop Peer-to-Peer Systems, 2002.
15. J. Skene, A. Skene, J. Crampton, and W. Emmerich, “The Monitorability of Service-Level Agreements for Application-Service Provision,” Proc. Sixth Int’l Workshop Software and Performanc, pp. 3-14, 2007.
16. BelGOnet,“Service Level Agreement,”http: //www .belgonet.com/website/UK/Service_Level_Agreement_UK.pdf, 2008.