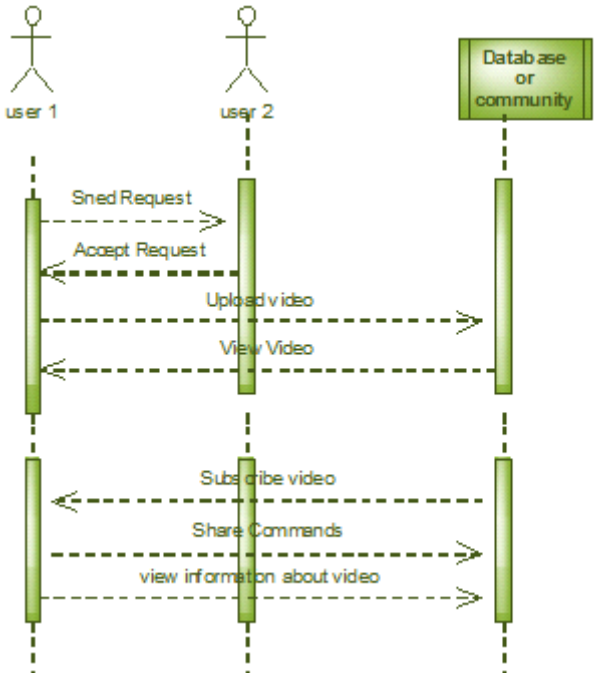

Figure 8.5: Sequence Diagram

9. System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

9.1 Types of tests

9.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

9.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

9.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- **Valid Input:** identified classes of valid input must be accepted.
- **Invalid Input:** identified classes of invalid input must be rejected.
- **Functions:** identified functions must be exercised.
- **Output:** identified classes of application outputs must be exercised.
- **Systems/Procedures**: interfacing systems or procedures must be invoked.

9.1.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

9.1.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

9.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

9.1.7 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

9.1.8 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

9.1.9 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements

10. Results and Discussion

The following are the results obtained of our work. The figure 10(a) represents the sign up page of the new user. The figure 10(b) represents how to upload image for the profile picture. The figure 10(c) represents the friends list. The figure 10(d) represents how to comment on uploaded image. The figure 10(e) represents how to search the video. The figure 10(f) represents how to upload video. The figure 10(g) represents how to play uploaded video. The figure 10(h) represents how to comment on uploaded video.

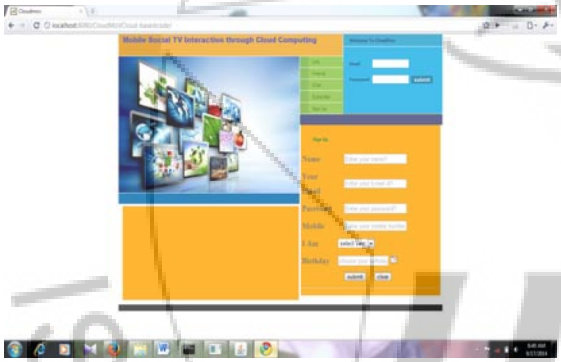

Figure 10(a)

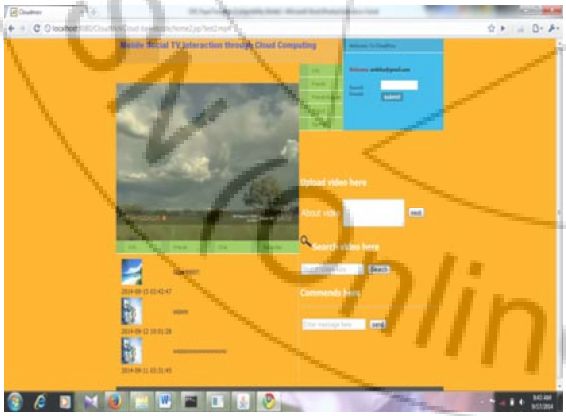

Figure 10(b)

Figure 10(c)

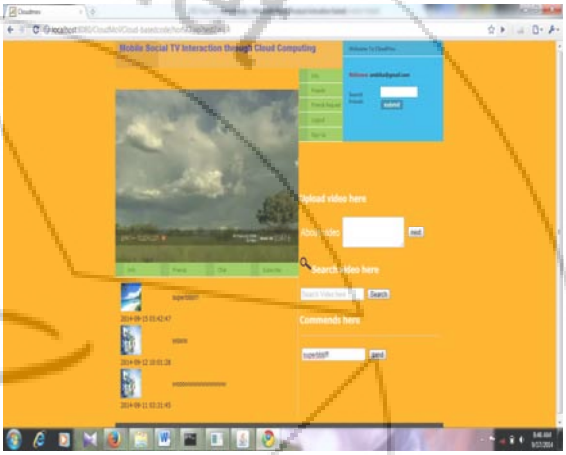

Figure 10(d)

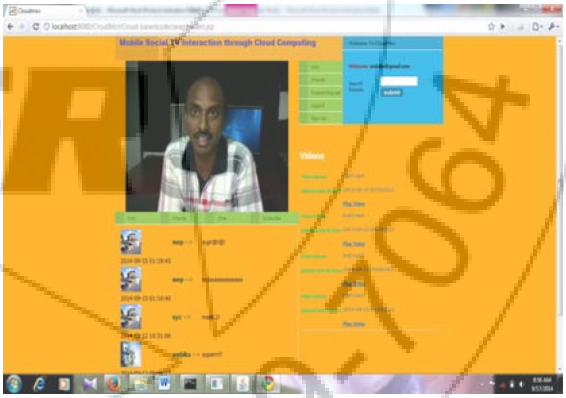

Figure 10(e)

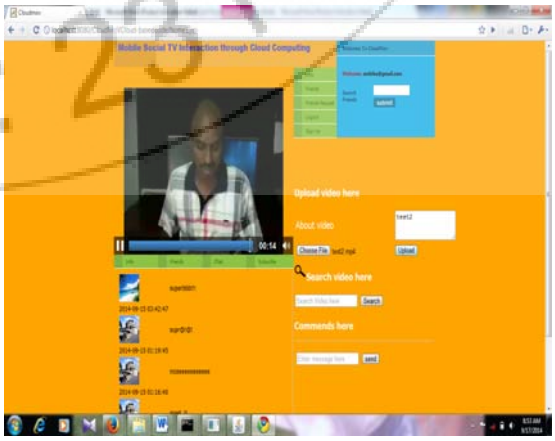

Figure 10(f)

Figure 10(g)

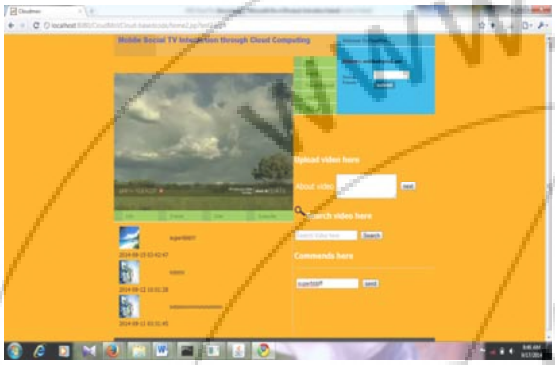

Figure 10(h)

11. Conclusion

We conclude results prove the superior performance of CloudMov, in terms of transcoding efficiency, timely social interaction, and scalability. Here social mobile users can import a live or on-demand video to watch from any video streaming site, invite their friends to watch the video concurrently, and chat with their friends while enjoying the video and even u can search the video and can comment on the video.

12. Future Enhancement

In the current prototype, we do not enable sharing of encoded streams (in the same format/bit rate) among surrogates of different users. In our future work, such sharing can be enabled and carried out in a peer-to-peer fashion, e.g., the surrogate of a newly joined user may fetch the transcoded streams directly from other surrogates, if they are encoded in the format/bit rate that the new user wants.

References

[1] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," IEEE Pervasive Computing, vol. 8, pp. 14–23, 2009.
[2] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in Proc. of IEEE INFOCOM, 2012.
[3] Z. Huang, C. Mei, L. E. Li, and T. Woo, "Cloudstream: Delivering high-quality streaming videos through a cloud-based svc proxy," in INFOCOM'11, 2011, pp. 201 205.
[4] T. Coppens, L. Trappeniners, and M. Godon, "AmigoTV: towards a social TV experience," in Proc. of EuroITV, 2004.
[5] N. Ducheneaut, R. J. Moore, L. Oehlberg, J. D. Thornton, and E. Nickell, "Social TV: Designing for Distributed, Sociable Television Viewing," International Journal of Human-Computer Interaction, vol. 24, no. 2, pp. 136–154, 2008.
[6] A. Carroll and G. Heiser, "An analysis of power consumption in as smartphone," in Proc. of USENIXATC, 2010.
[7] What is 100% Pure Java, http://www.javacoffeebreak.com/faq/faq0006.html.
[8] J. Santos, D. Gomes, S. Sargento, R. L. Aguiar, N. Baker, M. Zafar, and A. Ikram, "Multicast/broadcast network convergence in next generation mobile networks," Comput. Netw., vol. 52, pp. 228–247, January 2008.
[9] K. Chorianopoulos and G. Lekakos, "Introduction to social tv: Enhancing the shared experience with interactive tv," International Journal of Human-Computer Interaction, vol. 24, no. 2, pp. 113–120, 2008.
[10] M. Chuah, "Reality instant messaging: injecting a dose of reality into online chat," in CHI '03 extended abstracts on Human factors in computing systems, ser. CHI EA '03, 2003, pp. 926–927.
[11] R. Schatz, S. Wagner, S. Egger, and N. Jordan, "Mobile TV becomes Social - Integrating Content with Communications," in Proc. of ITI, 2007.
[12] R. Schatz and S. Egger, "Social Interaction Features for Mobile TV Services," in Proc. of 2008 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, 2008.
[13] J. Flinn and M. Satyanarayanan, "Energy-aware adaptation for mobile applications," in Proceedings of the seventeenth ACM symposium on Operating systems principles, ser. SOSP '99, 1999, pp. 48–63.
[14] W. Yuan and K. Nahrstedt, "Energy-efficient soft real-time cpu scheduling for mobile multimedia systems," in Proceedings of the nineteenth ACM symposium on Operating systems principles, ser. SOSP '03, 2003, pp. 149–163

Author Profile

Ambika received Bachelor of Engineering degree from Poojya Doddappa Appa College of Engineering Gulbarga Karnataka India affiliated to VTU Belgaum and pursuing Master of Technology (M.Tech) (2012-2014) in Computer science and Engineering from Arjun College of Technology and Sciences Hyderabad India affiliated to JNTU Hyderabad India. Cloud Computing is the field of interest.

Ramya Bathula is presently working as Associate Professor in Department of Computer Science and Engineering, Arjun College of Technology And Sciences, Batasingram, Hayath nagar, R. R. Dist. She received her B. Tech degree in Computer Science and Engineering from Swami Ramanand Thirde Institute of Science & Technology in 2008, M. Tech degree in Information Technology from ASTRA Aurora Scientific Technology & Research & Academy in 2010.