- A lot of users processing power for compiling and executing the executable will be saved.
- Users workspace on cloud can be accessible from anywhere the user wants using Internet connection.
- This workspace or cloud storage space can be extended on request.
- The workspace is implemented with the high level of security algorithms which will prevent attackers from intruding in the system.

4. Proposed System Architecture

The system is designed to work for three fields, we call it as zones. So, the system is divided in specially three zones:

1) Application Zone
2) Communication Zone
3) Database Zone

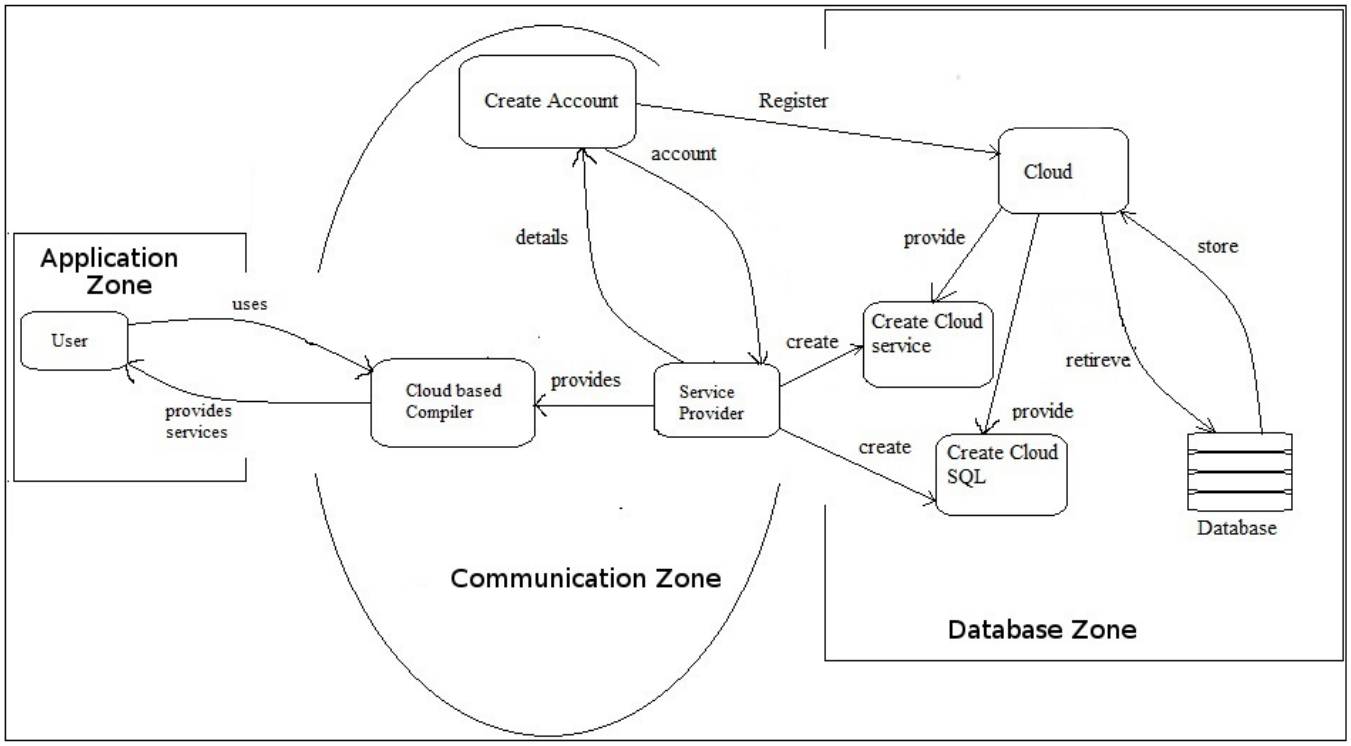

Figure 3: Architecture of the System

Application Zone: The application zone consists of the interface from where client can interact with the proposed system. The modules included in this zone are the Android application and the browser. The Android application is only for versions Gingerbread and above. Any person who doesn't have Android can also use the proposed system through web browser. This web browser can be for any platform like Windows, Linux, Mac OS, etc. These interfaces will provide the user with editors and various options through which user can avail functions needed such as compile and file upload. The application zones must be provided with an Internet connection. Without Internet connection, the compilers can't be used. Only editors can be used for writing the code and can be stored. Whenever the device gets Internet connection, the files will be uploaded automatically if the user wants.

Communication Zone: After once code being sent for compilation, the flow will move into the communication zone. The communication zone is the core part of the model. First, the code's language is used to detect the type of language the code consists. So that the code should be sent to appropriate compiler. Communication zone also includes scheduling the compilation queue and to check whether the compiler is idle or not, if not then the code goes to wait state. After the compiler being detected in the idle state, the codes are sent directly for execution. For getting access to the workspace, the user has to register first time and then login. This transferring of user-name and passwords in an encrypted format is included under communication zone.

Database Zone: The Database zone consists of total back-end contents such as workspace, user-name and passwords. These passwords are saved in encrypted format in the tabular database. The users will be provided with the limited workspace for storing their codes or projects. Whenever any particular user logs in, he will be provided with his workspace only. These files are accessible either from Android application or from the web browser.

5. Modeling Diagrams of System

The whole system's data flow can be well understood from these data flow diagrams. The system can be explained well only in splitting at two levels.

DFD Level 0: Level 0 states in general what the system is all about. User will send code to cloud compiler and the compiler will send back the output to the user.

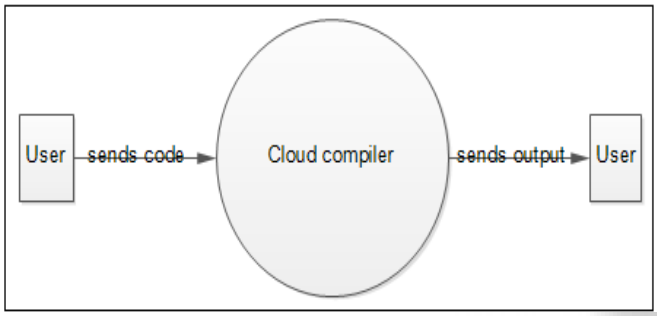

Figure 4: DFD Level 0

DFD Level 1: Level 1 goes little deeper in the DFD. It shows the two interfaces through which user can interact. The interfaces are web browser and the Android application. Then they interact with cloud for compilers and database. The compilers can interact only if they are in Free State.

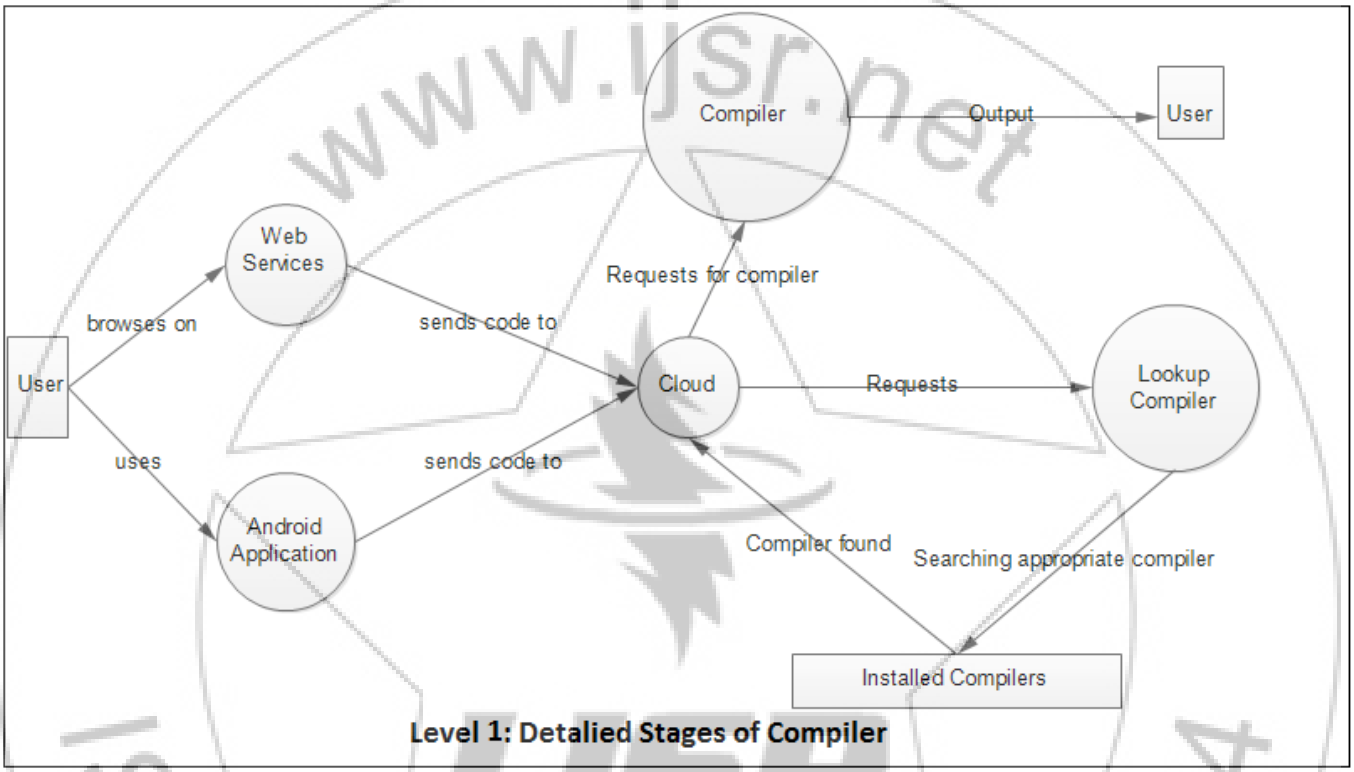

Figure 5: DFD Level 1

6. Future Scope

Although Cloud Compiler has many advantages as compared to its limitations, the future work on this system can include a lot of additional features which will help the user for using this system more interactively. The proposed system's one of the internal goal is to create a system, that will be so user friendly such that user will be most comfortable for programming in this system. For achieving this goal some more work has to be done in future such as:

1. Availability of few more compilers: Currently the system contains C, C#, Java compiler. Later on, C++, PHP, PYTHON, FORTRAN, COBOL support can be added.
2. Parallel compiling for more numbers of codes at a time can be setup with the help of load balancing mechanisms in the cloud. This can form parallel queues hence will decrease waiting time.
3. Implementation of proper real-time input system can be included in the proposed type of systems. Currently, the feature of accepting runtime inputs in the system is not present.
4. Due to such incompatibility with the run-time interaction with the system it creates the high level of complications to provide a debugger in the system.
5. Further, systems can provide the public forums to discuss the problems faced during the development of their projects. These discussions should be accessible all over the web.
6. One interesting thing that can be included in the system is that, user should be able to share their workspace with whom they want. This sharing can be for a person, a group or public with the proper privileges and access rights facility.

7. Conclusion

The online cloud compiler can be a future for compilers. Cloud is a concept like a boon for today's generation. Because of compiler being on cloud general users won't have to indulge in installation or upgrading of compilers. Using a compiler from the low end devices is the great advantage a programmer can have. Users who do not have Android can also grab the opportunity of using this system through web browser. A private workspace is the cherry on

cake for the registered users from where they get a storage area for codes and executable. Such list-full advantages and many more can make the cloud compiler for multiple languages, a successful system.

References

[1] Mayank Patel, Online Java Compiler Using Cloud Computing, International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 2, issue 2, January, 2013.
[2] Namrata Raut, Darshana Parab, Shephali Sontakke, Sukanya Hanagandi, "Cloud Documentation and Centralized Compiler for Java & Php", International Journal Of Computational Engineering Research (ijceronline.com) vol. 3 issue. 3, pg. 17, March 2013.
[3] Sajid Abdulla, Srinivasan Iyer, Sanjay Kutty, "Cloud Based Compiler", International Journal of Students Research in Technology and Management, vol. 1(3), pg. 308, May 2013.
[4] Palak Makhija, Naveen Hemarjani, "Implementing SAAS: Cloud Computing and Android Based Application Framework for C Programming", IOSR Journal of Computer Engineering (IOSR-JCE) vol. 11, issue 5, pg. 74-78, May. - Jun. 2013.
[5] Mehare Suraj, Paliwal Poonam, Pardeshi Mangesh, Begum Shahnaz, "Private Cloud Implementation for Centralized Compilation", International Journal of Soft Computing and Engineering (IJSCE), vol. 3, issue 5, November 2013.
[6] Anirban Kundu, Chandan Banerjee, Rana Dattagupta, "SaaS Oriented Generic Cloud Compiler", International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013.
[7] Yogesh Bhanushali, Dwij Mistry, Shraddha Nakil and Sharmila Gaikwad, "Object Oriented Analysis of Centralized C# Compiler Using Cloud Computing with UML", International Journal of Latest Trends in Engineering and Technology, pg. 268, vol. 3, issue 3 January 2014.
[8] Kavita M Garawad, Girish S., ".NET Compiler using Cloud Computing", vol. 3, issue 5, pg. 546, IJRET: International Journal of Research in Engineering and Technology, May 2014.