Initial Approach Report

1. Indicate your team name, names of team members and, which sub-project you are working on.
A: Group - 09; Ankit Bahuguna and Zeeshan Nasir; VOIP Module

2. Which programming language, operating system you want to use and the reasons for choosing them.
A: JAVA and Linux (ubuntu & windows for testing), because of our past working experience and our common interest in exploring the java based sound manipulation libraries. Also, Eclipse will be used as our development environment.

3. What type of build system is used.
A: Maven

4. What measures do you intend to take to guarantee quality of your software. • How do you write test cases for your software • Quality control. For example: by using Valgrind, cppcheck, etc.
A: We will try to think about several scenarios where the application might crash. Some of the test cases include:

- Two client configuration to check the working of VOIP Module.

We also intend to measure and improve the performance of our application.

We intend to investigate the following tools for maintaining/guaranteeing the quality of our application:

- **JProfiler:** http://www.ej-technologies.com/products/jprofiler/overview.html or,
- **Eclipse Java Profiler:** http://eclipse.org/tptp/home/documents/tutorials/profilingtool/profilingexample_32.html

5. What libraries are already available to assist you in your project.
A: We will be using/ exploring the following libraries and packages:

1. Interaction with OS sound Modules: JAVA media framework (http://www.oracle.com/technetwork/articles/javase/index-jsp-140239.html)
2. Diffie Hellman / Pseudo Identity generation: Natively supported in Java for basic key exchange. (http://docs.oracle.com/javase/7/docs/api/javax/crypto/KeyAgreement.html)
3. RTP/ SRTP (jRTP: https://java.net/projects/jrtp)
4. Socket Programming: java.net.* (Native support in Java) http://docs.oracle.com/javase/7/docs/api/java/net/Socket.html
5. Apache commons logging (http://commons.apache.org/proper/commons-logging/)

6. Apache commons lang (https://commons.apache.org/proper/commons-lang/)
7. HPPC (High Performance Primitive Collections for Java) (http://labs.carrotsearch.com/hppc.html). We will be exploring this library in our case to check if there is indeed any performance benefits.

6. Which license do you intend to assign to your project's software and reasons for doing so.

A: Apache Licence 2. Free to distribute and to make incremental changes. Because it will help any future user of the module to integrate the same, make changes freely and add to his own modules for a fully working VOIP system.

7. Team members' previous programming experience which is relevant to this project

A: Both the team members are comfortable programming in JAVA with 3 years of working experience in the stated programming language. We also want to explore various design patterns taught in the TUM course "Patterns in Software Engineering" (taken by both), where Java was the primary language.

8. How do you plan to share the workload in your team

A: Sub-Modules / tasks in our project; Designated as **A: Ankit; Z: Zeeshan** -

1. **Front End:** Command Line Interface- **A**
2. Integration with **System Audio Interface** (ex. Pulse Audio) - **A/Z**
    a. Receive Audio Stream
    b. Record local microphone Input Stream.
3. **Call-Control - Z**
    a. Initiation Calls (Interface)
    b. Receiving Call (Interface)
4. **Data-Control - A**
    a. Audio Stream Control
    b. RTP / SRTP Control
5. **Crypto-Management - Z**
    a. Session key generator
    b. Diffie-Hellmann keys generator
    c. SHA256 keys generator - **A**
6. **DHT Manager**: PUT / GET requests on DHT for calle. - **A**
7. **Pseudo Identity Manager - Z**
    a. Create pseudo identity
8. **Cache Manager:** Caches a pseudo-identities of previously performed DHT GET. - **Z**
9. **Logging Manager** / Error Reporter - **A/Z**
10. **KX Manager** - **Z**
11. **Attack Prevention Manager** - **A/Z**
    a. Performs Self calling
12. **Other Investigations**: **A/Z**
    a. Multithreading

b. Performance Issues (Call Dropping / Packet Dropping etc.)
c. Security aspects.

9. Issues and complaints

A: We propose the following suggestions / questions:

- We want to apply Agile software development methodology, so that we could continuously get the feedback from your side and get some suggestions.Sessions can be conducted but the tutors to examine the progress and current working state of the project, whether or not we are on the right track and suggest improvements based on it.
- How are we supposed to test our implementation? Will there ever be any testing module provided or test specifications given which can help us to recognize errors in our final system?
- Can a timeline be defined which tells us about the date of integration of our module with others to check the working of whole system, so that we can work on the issues post integration and rectify them?