



## LG Software Architectures Training Program - Project Description

### Voice over IP (VoIP) and Video Communication System

#### Project Overview

The goal of this project is to design a robust and secure VoIP and video communication system that consists of a Microsoft Windows application and a backend server that provides various capabilities. The project sponsor, SecureMeet Inc., wants to launch this new product. Key motivators for this project are:

- SecureMeet Inc. will select the best architecture/solution based on the results of this competition among the teams in this course.
- Ensure quality voice and video communication that is also secure and reliable.
- Ensure that the system backend server and user application minimizes network bandwidth as necessary to maintain communication between the parties.
- Provides the ability to easily add or replace codecs and/or encryption algorithms.

In this project, each team will create a software architecture and then develop and build the software for a video enabled VoIP communication system that runs on a Microsoft Windows laptop and a backend server that provides user registration, contact identification assignment, directory services, and functionality to support video conference calls. During a video call, the video of the person speaking should be displayed to all participants.

All teams will interact, communicate their client related questions or concerns with a single SecureMeet point of contact.

**Please note that teams are NOT allowed to use any Session Initiation Protocols (SIP) or any VoIP libraries in source code or library form.**

#### System Description

The goal of this project is to design and implement a software system for a video and Voice over IP (VoIP) communication system that provides the following functionality.

##### *Microsoft Windows User Application*

The Microsoft Windows user application shall provide the following functionality:

1. The ability for the user to register with the system.

- a. The user shall provide the system their e-mail address and password. The system shall ensure that the user's password is secure.
- b. The ability for the user to update their e-mail address and password.
- c. The ability for the user to change their password or reset it in the event it is lost.
- d. After successful registration the system shall assign the user a unique contact identification name (contact identifier) this can be the users e-mail address or some other name chosen by the user if it does not conflict with other user's contact identifiers already in the system.
2. The ability to request a conference call identifier (like a contact identifier) for a particular date, time and duration from the server that is then distributed to participants via user application in an upcoming conference meeting list or via e-mail.
3. The ability to create a contacts list. To add a contact to the list the user can search users by last name, first name, address, e-mail, or contact identifier. When a contact is associated with a contact identifier the VoIP application shall display the contact's name instead of the contact identifier.
4. All communications shall:
  - a. Provide high quality low latency with strategies for dealing with packet or data loss and jitter.
  - b. Support a video call if there is enough available bandwidth.
  - c. Adjust bandwidth use as necessary to maintain communication quality.
  - d. Provide secure communication.
  - e. Provide reliable signaling for both inbound and outbound calls including dropped call detection.
  - f. Notify users of lost connections.
  - g. Provide acoustic echo cancellation (AEC).
  - h. Provide voice activation detection (VAD). The user application shall not send voice data during periods of silence when the user is not speaking.
5. The ability to initiate a call using a contact identifier or the contacts list. During the call initiation, the user shall be presented with call status and outcome (answered, busy or rejected). During call initiation the user shall have the ability to end the call at any time.
6. Provide the ability to accept or reject calls while not in a call. Application shall show the caller's contact identifier or contact name during an incoming call.
7. Provide the ability to terminate a call at any time while in a call. If a call is terminated by one user, the other caller shall be notified.
8. Notify the directory services component of the backend server of IP changes.
9. Provide the ability for the developer/maintainer to easily add or replace codecs and/or encryption algorithms.
10. Application shall be brought to the foreground during an incoming call.

### *Server Application*

The server application shall provide the following functionality:

1. Register each user's e-mail address and password. The system shall ensure that the user's password is secure.
2. The server shall provide a directory services capability that maps a device's contact identifier to its current IP address.
3. Ability to schedule and support video conference calls. Conference calls shall be scheduled with participants via a conference identifier.

4. The ability for the administrator to view user accounts and disable, enable, and delete user accounts.
5. Provide the ability for the developer/maintainer to easily add or replace codecs and/or encryption algorithms.

## System Hardware

This section describes the hardware that will be utilized for this project.

### *LG Wi-Fi Network*

Teams will utilize LG's Wi-Fi network for normal application development, testing and demonstrations. If desired, LG may provide teams with individual Wi-Fi routers.

### *Test Router*

This router has two Wi-Fi access points (AP's) and supports the ability to add network jitter and randomly drop network packets at rates specified by the user. This router is controlled via ssh using the supplied shell scripts.

### *Team Laptops*

Teams will use their laptop to serve as part of their development environment and possibly to run the server and its user interface. Therefore, team laptops will need to be configured to connect to LG's Wi-Fi Network or a LG supplied Wi-Fi Router and the supplied test router.

## System Software

Microsoft Windows 10 or 11.

## Required Software

Microsoft Visual Studio Community 2022 (64-bit) Version 17.5.5 or later

## Sample Code Archive

An example Microsoft Windows VoIP Video application written using Microsoft Visual Studio Community 2022 is provided as shown in Figure 1. Please note this example application should NOT be considered as an exemplar architecture or code. It merely provides a means for demonstrating how some basic features could be implemented and should not be used as an architecture to build upon or extend. Some example code may also not be implemented in the best or most efficient way and may not be complete. The example application also does not contain any code to support voice quality, reliability, or security. Also, it does not rely on a directory server.

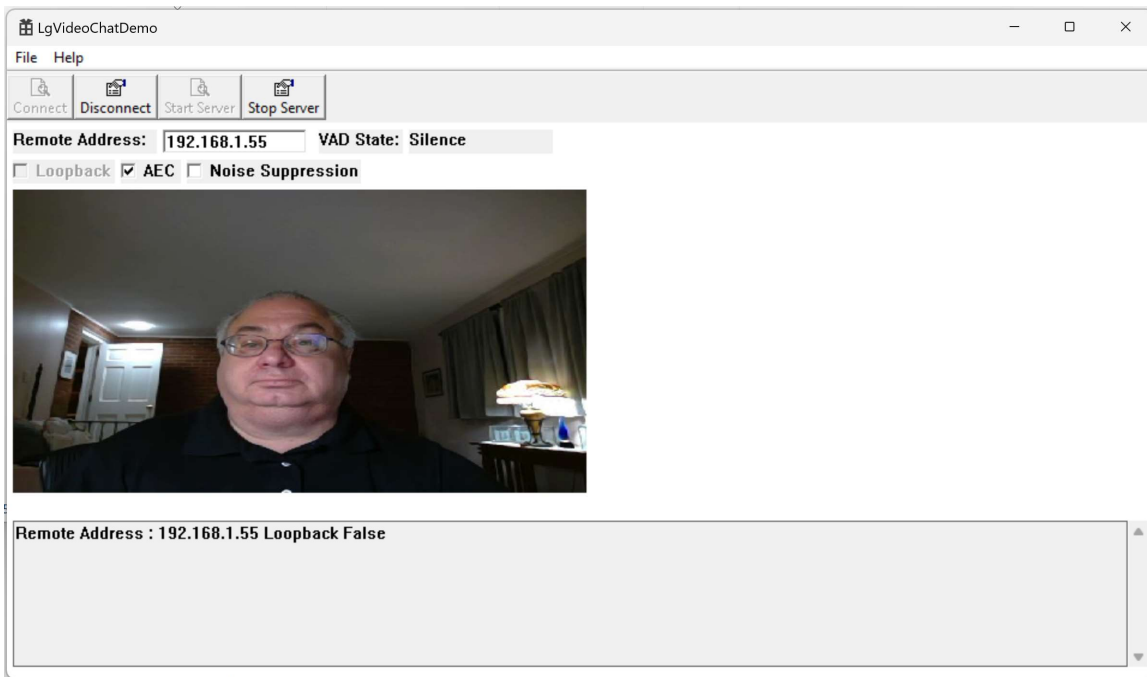


Figure 1: Sample VOIP Video Conferencing Application

## Project Deliverables

There are three milestones required for this project (each of which will be graded independently):

1. **Requirements, plan and risks:** For the first milestone the team will turn in the prioritized architectural drivers, any draft design decisions they are considering, the technical risks identified, and the experimentation plans (or any results already achieved).
2. **Experimentation results, design, plan for construction:** The team will turn in the results of experiments conducted, a design description with different architecture views, and the plan for construction.
3. **Demo and lessons learned:** For the final milestone, the team will demo the completed system, and present the lessons learned.

## Milestone 1

The submission can be informal documents describing the team's current understanding of the items listed below. Team mentors will be meeting with the teams to ask follow up questions.

- Project Plan
  - We will see if the design is sound enough to guide construction.
  - We will evaluate when the team will be able to determine how well the design supports the architectural drivers (based on the planned activities).
  - The plan should describe the overall architecture, the division of roles, the specific tasks planned, and the associated milestones.
- Architectural Drivers

- Are the QA requirements “actionable”? In other words, are they expressed in such a way that the team will be able to determine if a given design supports these drivers or not?
- Do the drivers seem to relate to the overall objectives of the project?
- Are the measures clearly derived from the overall goals of the project?
- Are the functional requirements understood?
- Is there a mechanism for prioritizing the requirements?
- Risk Assessment/Planned Experiments
  - What are the technical and non-technical risks? How do you assess each risk with respect to probability and impact in a H-M-L scale?
  - Are the open questions/issues clearly related to things that will affect the outcome of the project?
  - Have there been any actions identified to address the open questions/issues?
  - Are the experiments concretely articulated?
  - Is it clear what question/issue is being addressed by the experiments?
  - Will it be clear when the experiments are complete?
- Architectural Approaches
  - What is the overview-level description of the architecture?
  - What are the main architectural approaches (tactics, patterns, design strategies) in your solution?
  - Are the architectural approaches clearly related to the drivers (will they likely impact the properties of interest)?

## Milestone 2

Again, the submission can be informal documents describing the team’s current understanding of the items listed below. The mentors will be meeting with the teams to ask follow up questions.

- Project Plan
  - How has the plan changed?
  - Has the team been actively assessing risk and updating the plan accordingly?
  - Does the team have a plan for any remaining significant issues/risks
  - Does the team have a reasonable construction plan?
- Experiments/Results
  - What experiments have been conducted?
  - Have the results of the experiments addressed the open questions/issues?
  - What experiments remain?
  - Are the experiments focused on issues relevant to the overall goals of the system?
- Architecture
  - What is the architecture in terms of the organization of code units and their dependencies? (The team shall create a module view of the architecture.)
  - What is the architecture in terms of components and connectors (runtime perspective)? (The team shall create a C&C/runtime view of the architecture.)
  - What is the architecture in terms of the supporting infrastructure (deployment perspective)? (The team shall create a deployment view highlighting component allocation to hardware elements and communication channels.)
  - Have the experiments led to a refinement of the architecture?
  - Does the team understand the architectural approaches they have/will realize?
  - Do the architectural approaches align with the goals of the system?
  - Are there significant concerns that have not been addressed?
  - Has the architecture been evaluated?

## Milestone 3 - Final Demo and Presentation

For the final deliverable there will be both a team presentation and a demonstration of the final system.

### *Presentation (150 points)*

The presentation should cover:

- Quality attribute requirements for the system.
  - Present evidence that the team's performance and reliability approach is sound.
  - Present evidence that the team's security approach is sound.
  - Present evidence that the design/architecture can adjust to changing network conditions.
  - Present evidence how the developer/maintainer can easily add or replace codecs and/or encryption algorithms.
- Architecture description showing architecture views and highlighting key architectural approaches adopted and their rationale.
- Description and results of the architecture evaluation activities.
- Lessons learned (what went right, what went wrong, what would you have done differently).

### *Demo (350 points)*

There will be a live demo of the system. The criteria for evaluating the demo will be:

- User registration and contact identifier assignment: 20 points.
- Change user information: 20 points.
- Recover password: 20 points.
- Receive call with app in the foreground 20 points.
- Receive call with another app in foreground: 20 points.
- Receive call with application minimized: 20 points.
- Call initiation, incoming notification, and acceptance, rejection, user busy (answered, busy or rejected): 20 points.
- Lost connection detection: 10 points.
- Adding and modifying user contacts: 10 points
- Initiate call using contact identifier and contacts list: 10 points.
- Demonstrate video and call quality by counting for 30 seconds (echo cancel, clear voice): 60 points.
- Conference calling setup & conference calling with at least 4 participants (echo cancel, clear voice, video): 60 points.
- Demonstration of robustness in an adverse environment with packet loss, network jitter, and under network load (this includes voice and control data traffic): 60 points.
- Best UI: 40 points; 2nd Best UI: 20 points, 3rd Best UI: 10 points, 4th Best UI: 5 points (points will awarded based on client/sponsor judgment)

**Note: Any application crash or improper behavior during the demonstration will result in a 30-point deduction for each occurrence.**

## Assumptions and Hints

- The team may use any third-party software (except for SIP or complete solution VoIP libraries, this includes video conferencing). Teams must justify their choice of technology.

- Any programming language can be used.
- Assume that anything can fail... your design should account for failure and recovery to the greatest extent possible.
- Teams will use their own laptops for software development and to execute the user application and server software.