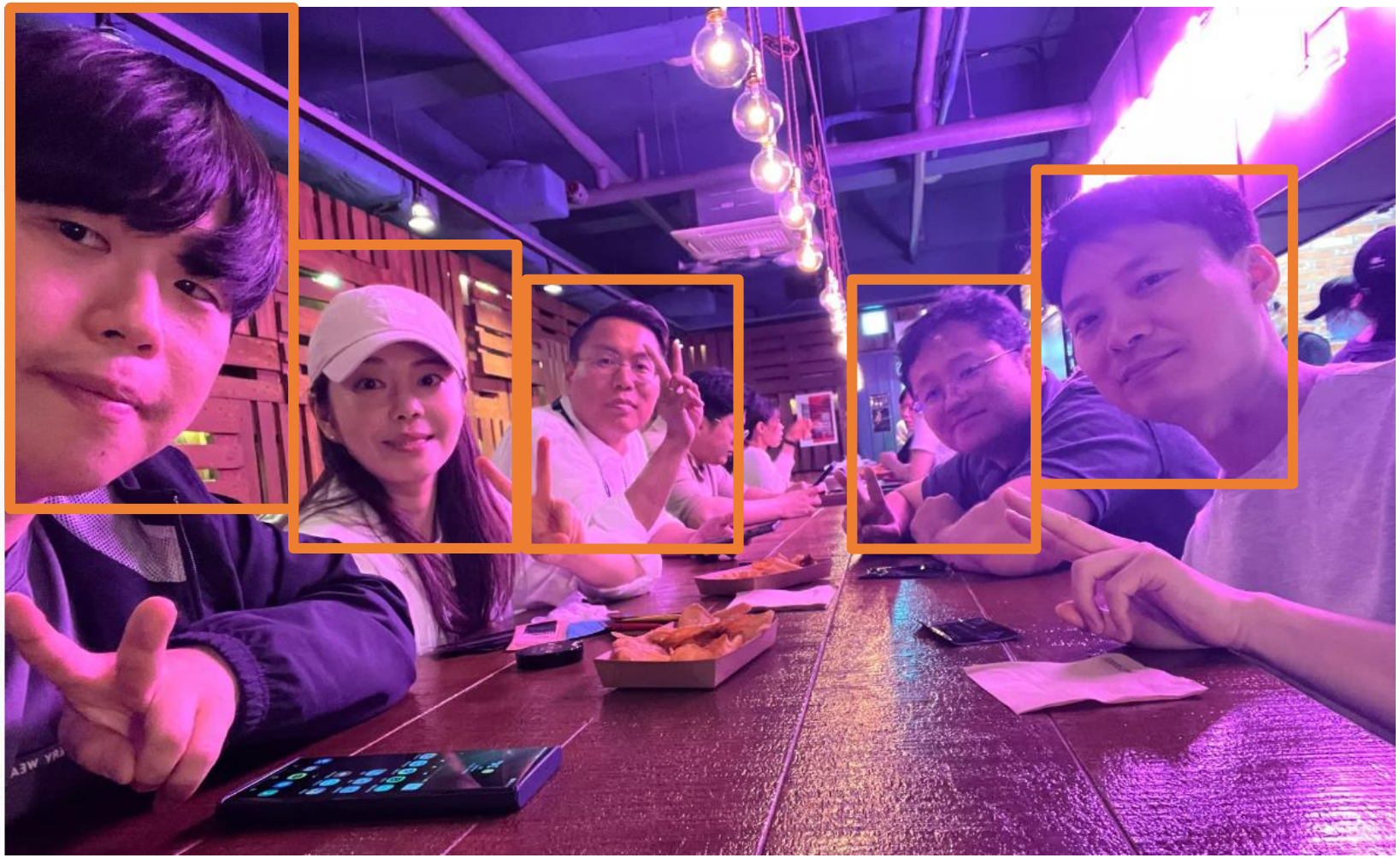
Phase 1 Presentation

Team 4 B1C2V3

Team introduction



Developer (Backend)
Truong Quang Viet

Our Wonderful Mentor
Clifford

TEAM Name : B1C2V3
1 member from BS company
2 members from CTO
3 members from VS company

Developer (App)
Hongjae Lim

Developer (App)
Minji Tae

Requirement & Test Manager
Youngjin Kim

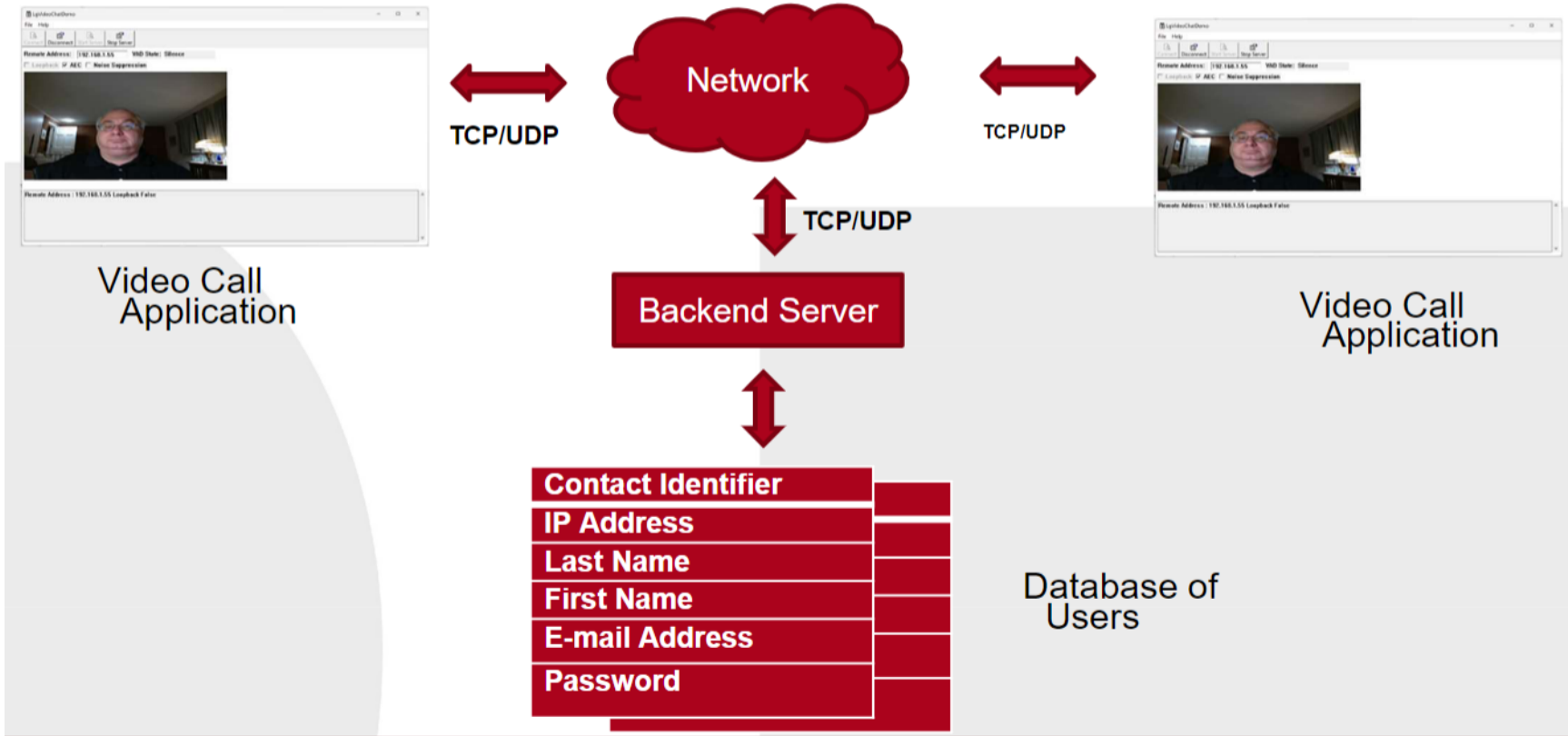
Threat Analyst & Architect
Chanhun Seung

Team leader
Jongoh Ha

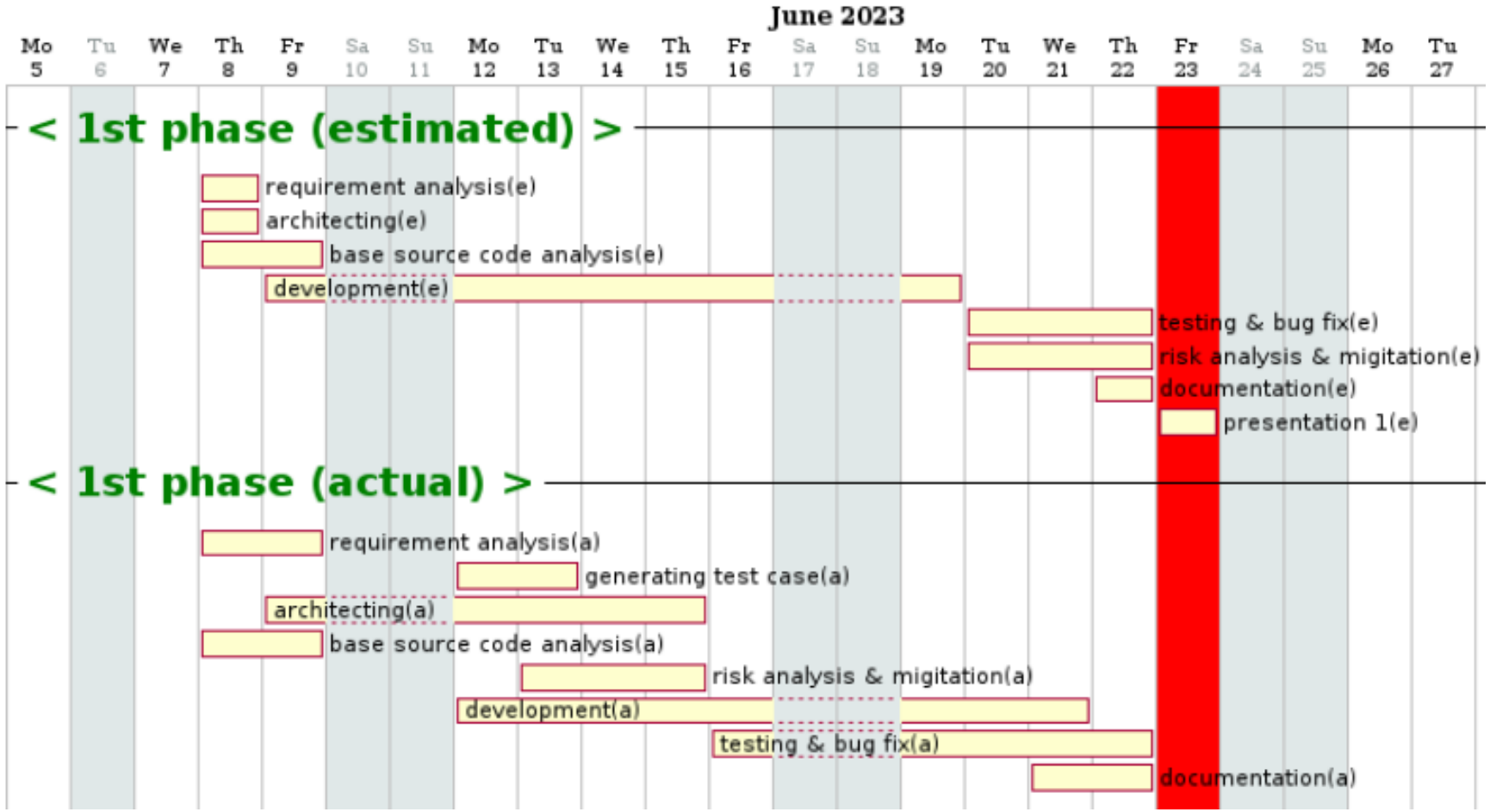
Project Overview

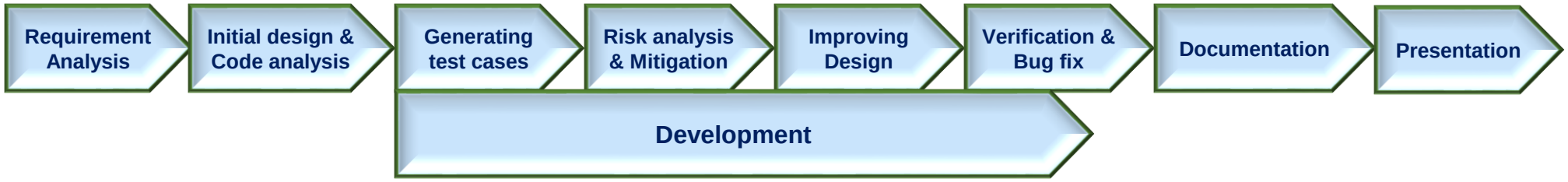
System overview

- ✓ Video Call Application for both business and personal users
- ✓ Video Call Communication over the Network
- ✓ User registration and login function with two factor authentication
- ✓ **Current design needs to be improved in terms of security**



Project Plan & Schedule





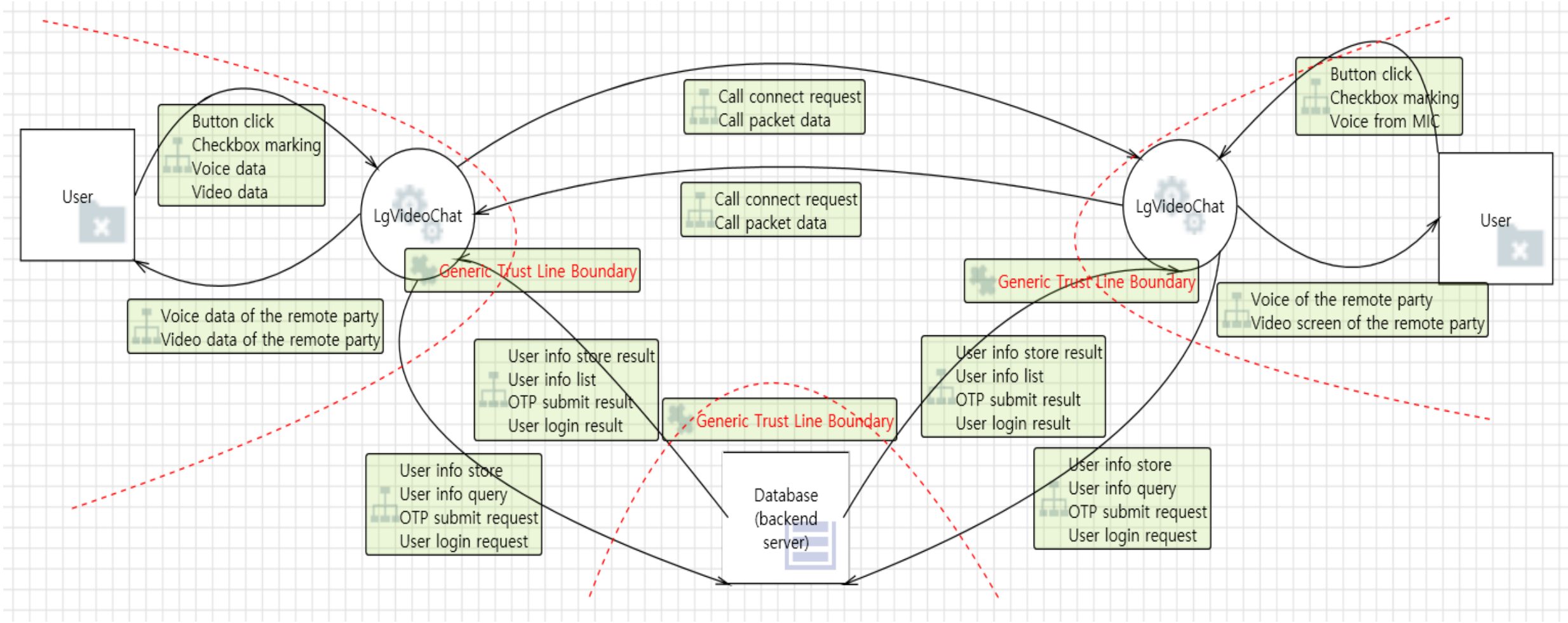
Requirement Analysis

Functional Requirement	New user registration	11
	Login	8
	User email update	8
	Periodic password reset	10
	Lockout due to incorrect password	13
	Reset password	7
	Unique ID	1
	Contact list	2
	Call	4
	Connection	2
	Notice	2
	Disconnect	2
	Activation	1
	Communication method	1
Non Functional Requirement	Performance	1
	Authentication	2
	Communication privacy	1
	Non repudiation	1
	Reliability	1
Total		78

- ✓ Number of Initial requirements : 19
  (functional : 14, non functional : 5)
- ✓ Analyzed initial requirement in **team workshop** and **mentor meeting**
- ✓ Requirement manager derived **78 system requirements** through additional analysis

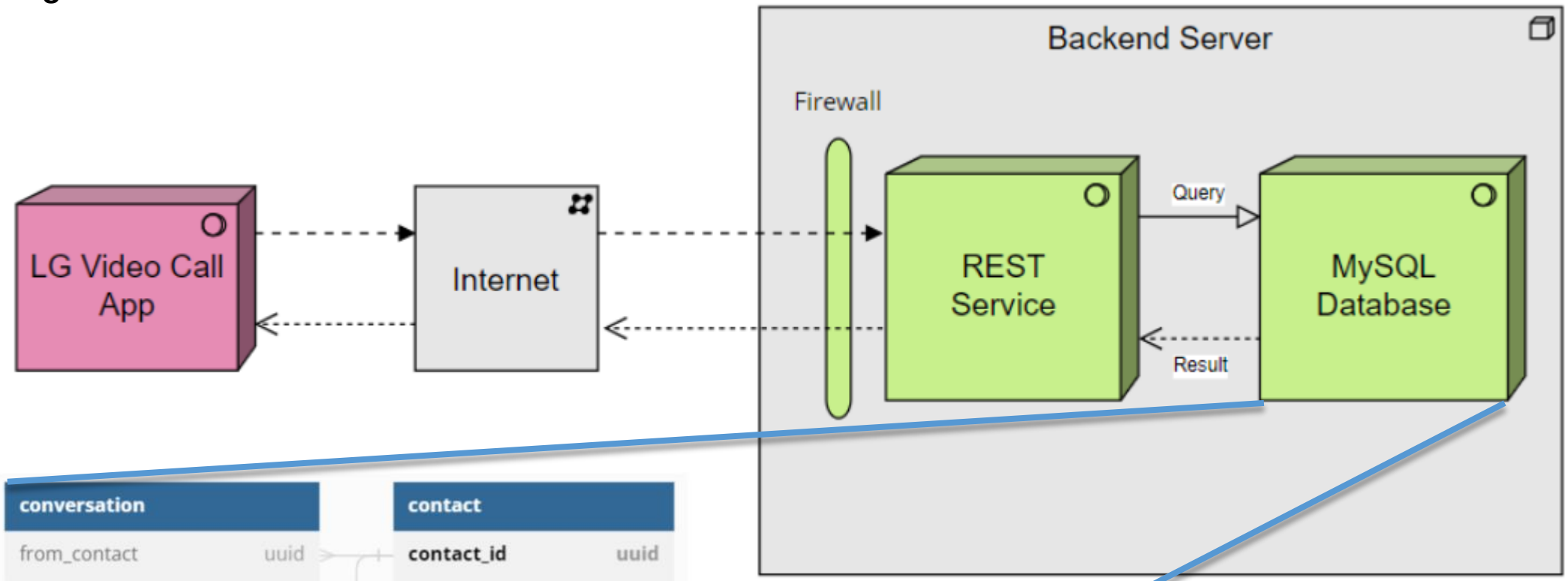
Initial Design (Overall System)

✓ **Initial Design reflecting all system entities and communication data**



Initial Design (Backend server)

- ✓ **Initial Design for backend server**

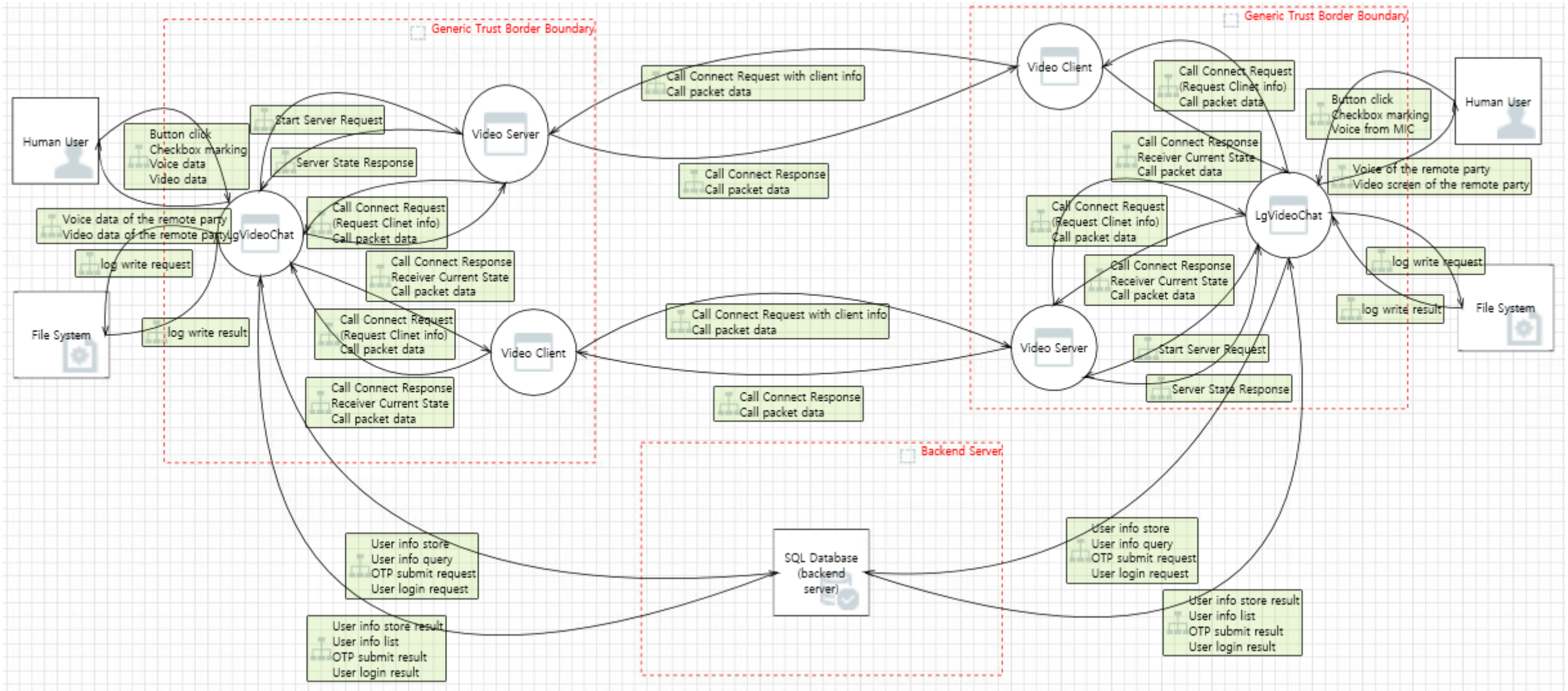


conversation	
from_contact	uuid
to_contact	uuid
started_at	timestamp
stopped_at	timestamp
status	integer

contact	
contact_id	uuid
email	varchar
first_name	varchar
last_name	varchar
password	varchar
ip_address	varchar

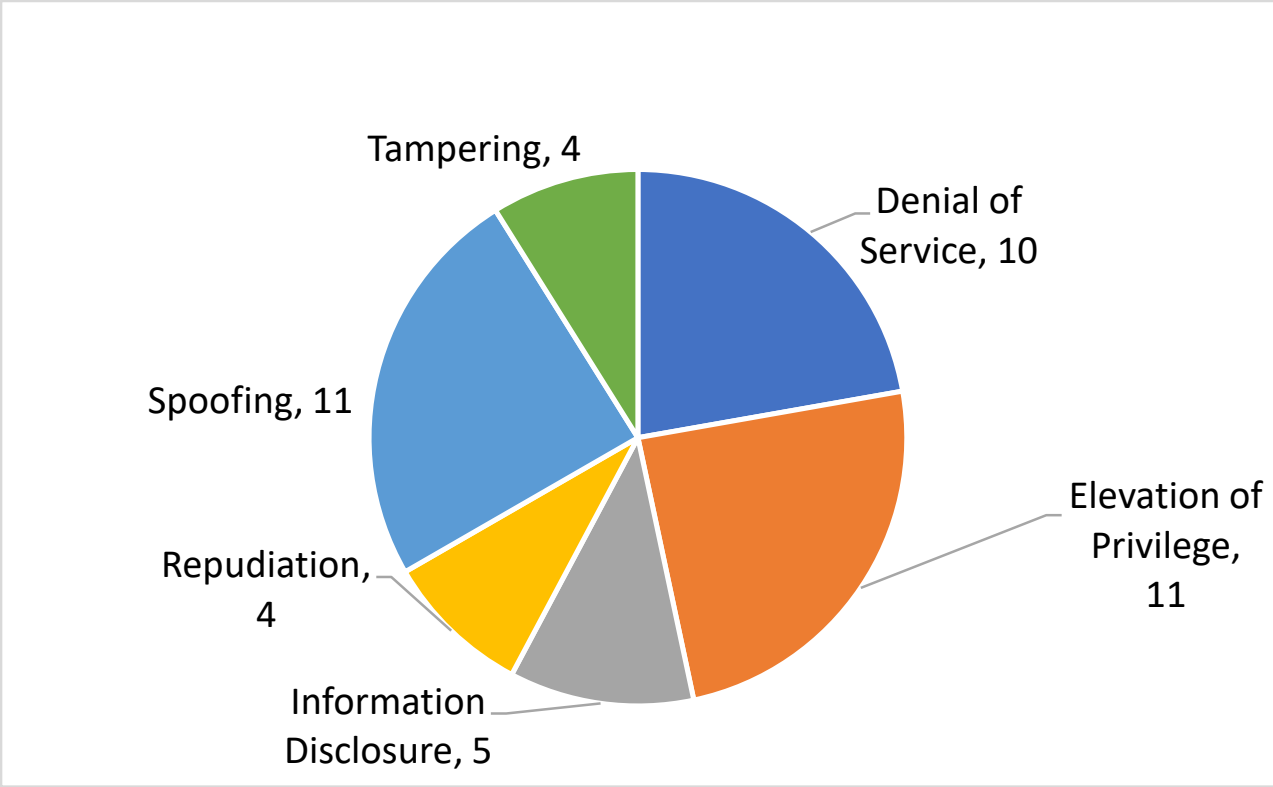
Threat Analysis

✓ **DFD and STRIDE methodology was used to perform Threat Analysis**



Security Requirements & Mitigations

Number of Identified threats : 45



- ✓ Prioritized each threat through team workshop
- ✓ Resulted in various score : 4, 6, 8, 10
- ✓ **Key Security Requirements** from high priority threats (Score 6, 8, 10) are as follows

1. PKI-based server authentication for App and Backend Server

2. Secure communication between Apps

3. Secure communication between App and Backend Server

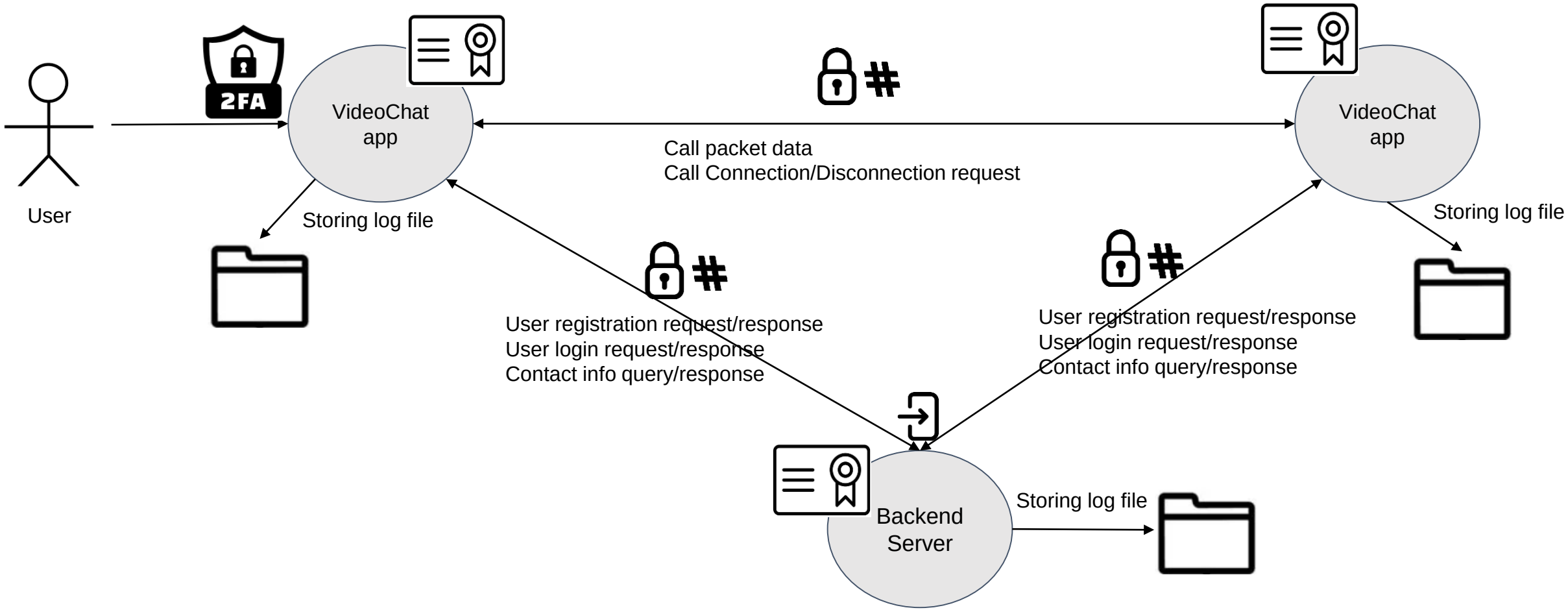
4. Two factor authentication using password and OTP to email

=> Initial requirement has been specified in detail

5. Input validation check by Backend Server

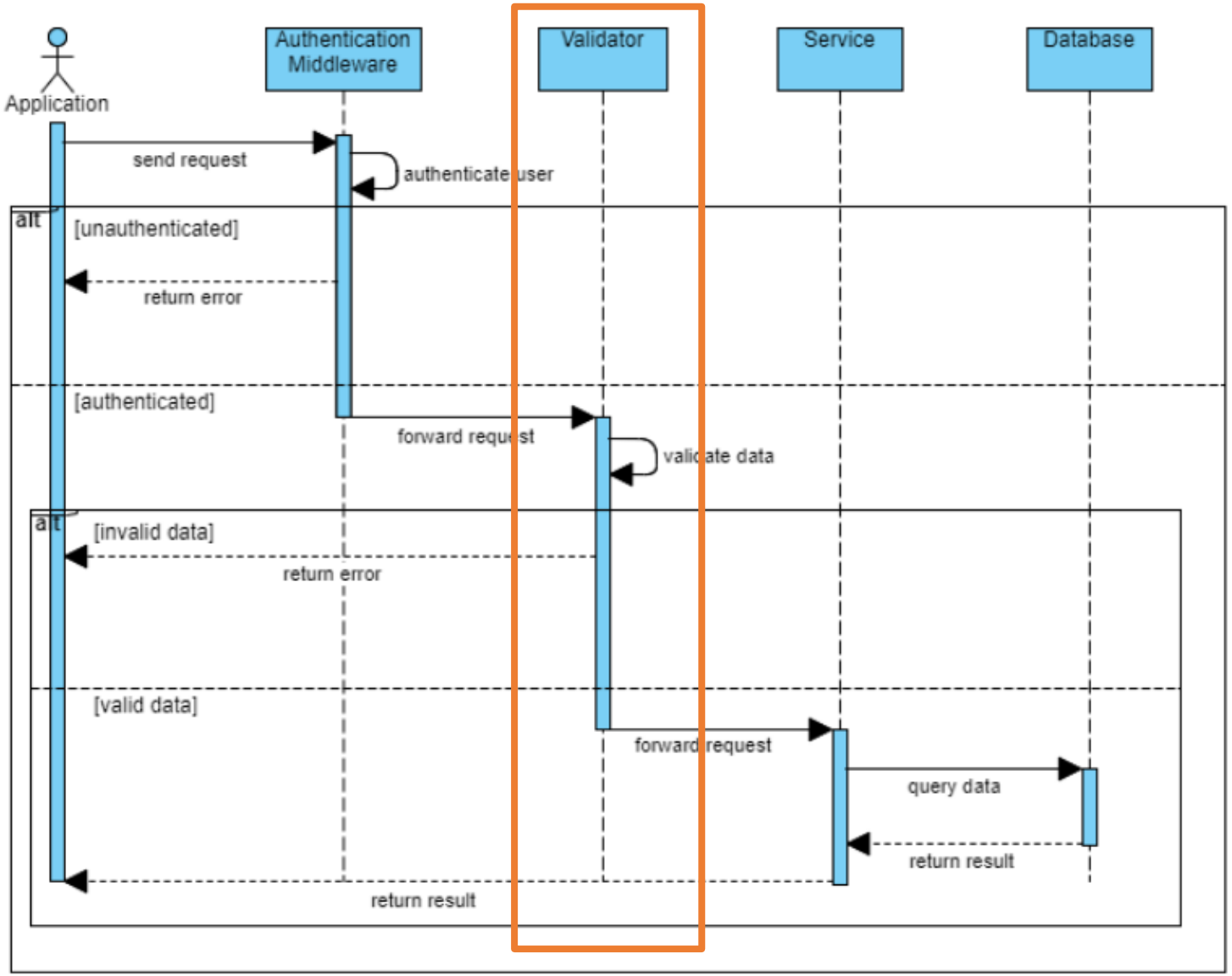
6. Storing log file by App and Backend server

Overall System Design for Security Requirements

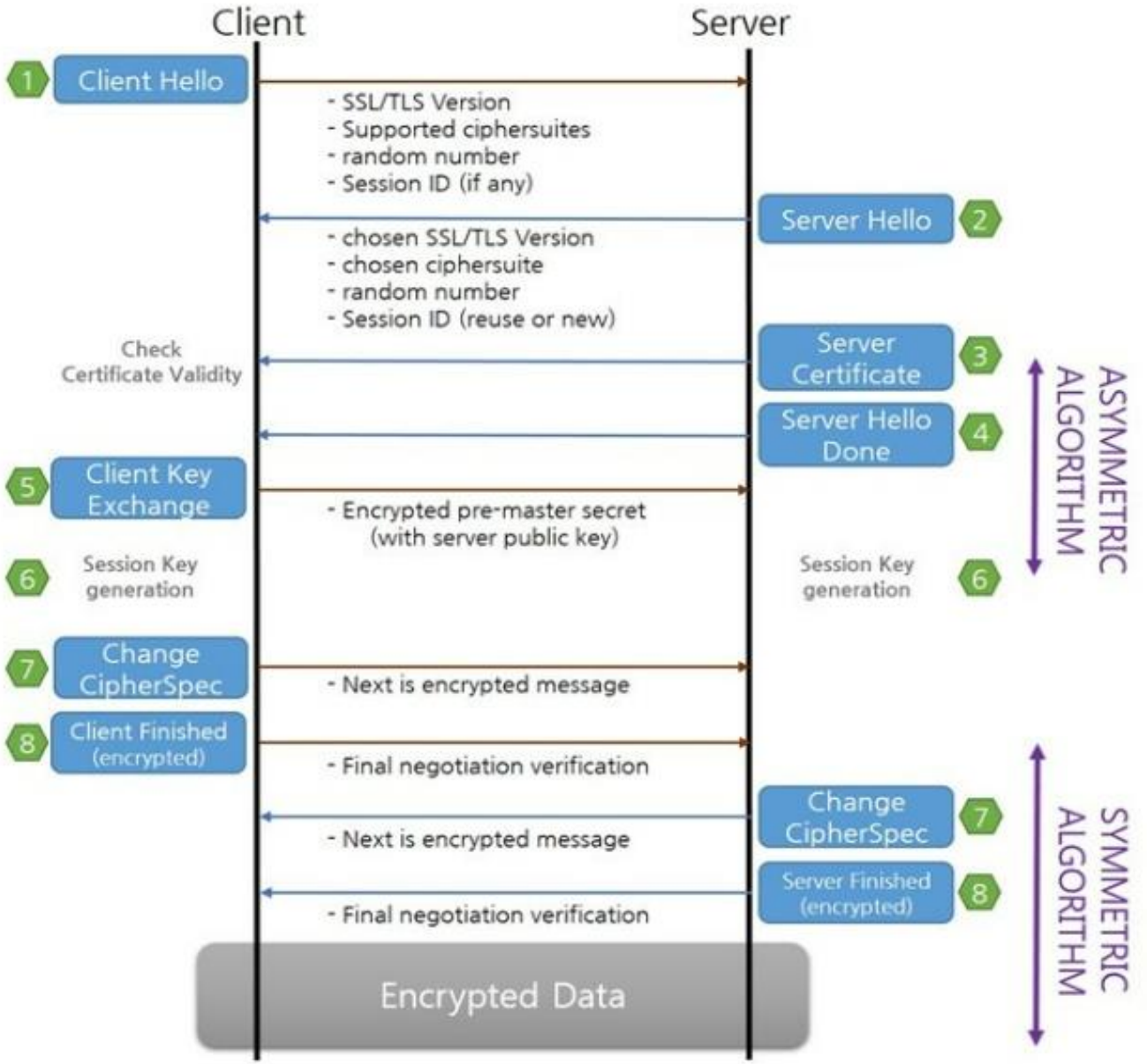


Sequence Diagram for Secure design (1)

- ✓ **Input Validation Check by Backend Server**



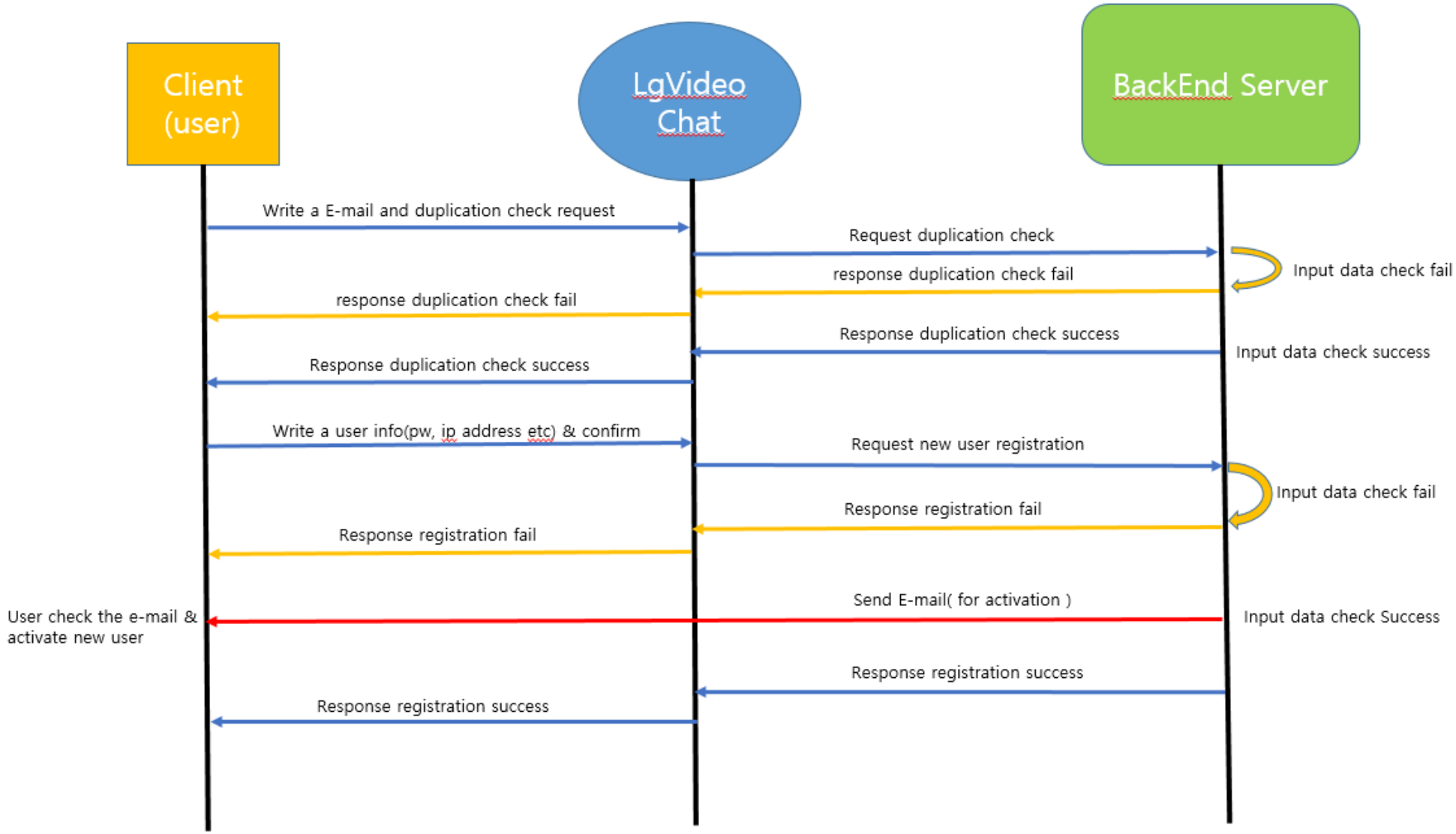
Sequence Diagram for Secure design (2)



- ✓ PKI-based server authentication for both Application and Backend Server
- ✓ Secure communication between Applications
- ✓ Secure communication between Application and Backend Server
- ✓ **Adopted Solution : TLS v1.3**

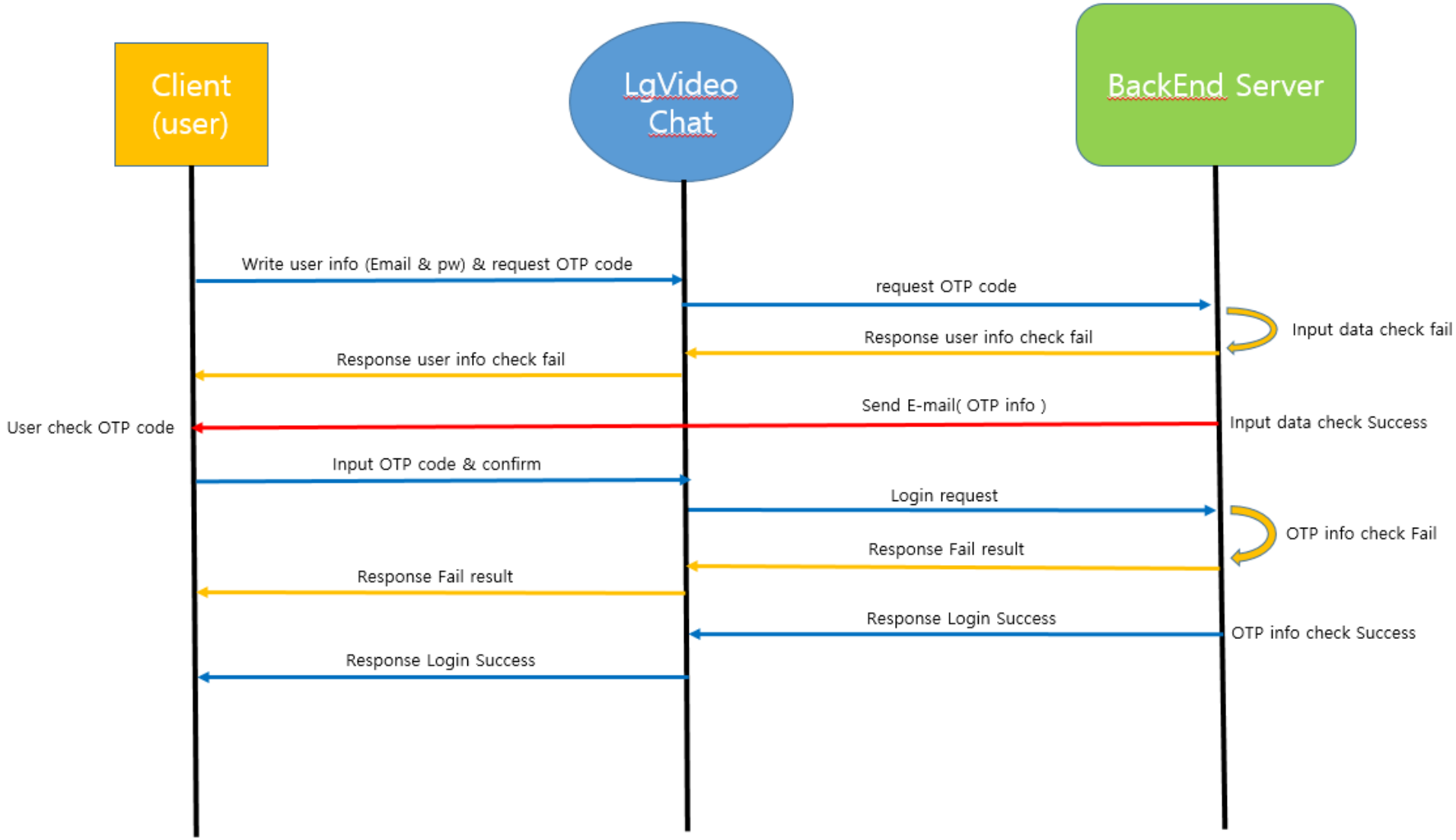
Sequence Diagram for Secure design (3)

✓ **Two factor authentication (User registration)**



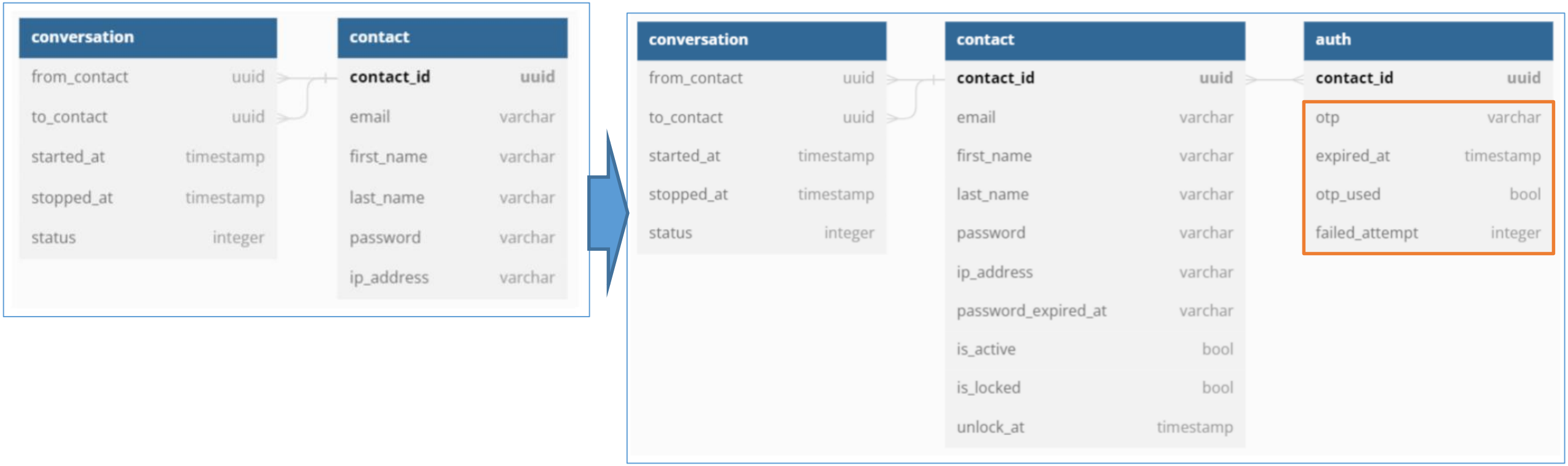
Sequence Diagram for Secure design (4)

✓ **Two factor authentication (User login)**



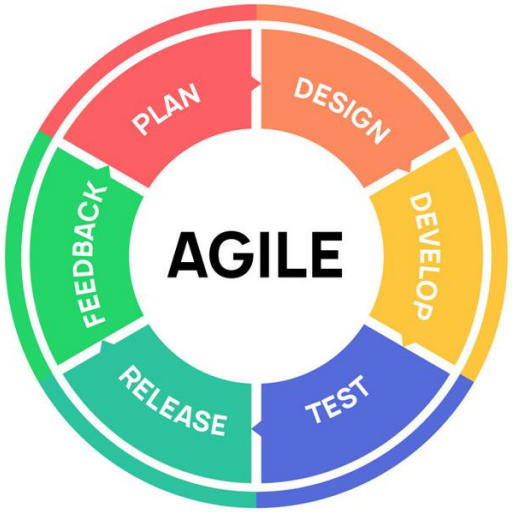
Database table for Secure design

- ✓ **Change in database table design for two factor authentication**



Implementation

- ✓ **Agile methodology was applied. The reason and how …**
  - Started implementation before completing all design due to lack of time
  - Need to reflect changed and added system design after threat analysis
  - Development and Verification was performed in parallel to find bugs earlier
  - Sync up meeting and sharing obstacles every day
    (http://collab.lge.com/main/display/SCSPECIALT/0.+Meeting+Minute)



- ✓ **Development environment and tools**
  - Visual Studio Community, Beyond compare
  - MySQL, Ethereal for Fake Email Service
  - Self signed certificate for server authentication
  - GitHub for sharing and integrating source code
  - Additional library : Openssl, Boost, Nlohmann-json for application

Verification

Test Case	Test Cycle1	Test Cycle2	Test Cycle3	Test Final
Sign-Up(9 → 6)	**PASS**(4) / **FAIL**(2)	**PASS**(4) / **FAIL**(2)	**PASS**(5) / **FAIL**(1)	**PASS**(6)
Sign-In(7)	**PASS**(2) / **FAIL**(5)	**PASS**(4) / **FAIL**(3)	**PASS**(4) / **FAIL**(3)	**PASS**(7)
Update(5)	**PASS**(2) / **FAIL**(2) / SKIP(1)	**PASS**(2) / **FAIL**(3)	**PASS**(2) / **FAIL**(3)	**PASS**(5)
Periodic P/W Reset(8)	N/A	N/A	**FAIL**(8)	**FAIL**(8)
Lockout due to an incorrect P/W(7)	N/A	N/A	**PASS**(3) / **FAIL**(4)	**PASS**(5) / **FAIL**(2)
~~**Reset P/W(7) : Optional Requirement**~~	~~N/A~~	~~N/A~~	~~N/A~~	~~N/A~~
Unique ID & Contact List (2)	N/A	N/A	**FAIL**(2)	**PASS**(2)
Call(4)	N/A	N/A	**PASS**(3) / **FAIL**(1)	**PASS**(3) / **FAIL**(1)
Connection, Notice and Disconnect(6)	N/A	N/A	**PASS**(3) / **FAIL**(2) / SKIP(1)	**PASS**(4) / **FAIL**(2)
Communication methods(2)	N/A	N/A	**PASS**(2)	**PASS**(2)
Total(~~57~~ → 47)	**PASS**(8) / **FAIL**(9) / SKIP(30)	**PASS**(10) / **FAIL**(9) / SKIP(30)	**PASS**(22) / **FAIL**(24) / SKIP(1)	**PASS**(34) / **FAIL**(13)
Pass Rate	**17%**	**21.2%**	**46.8%**	**72.3%**

Test case

Generated based on Functional requirements

Test purpose

- To verify initial functional requirements
- To verify additional security requirements

Test constraints

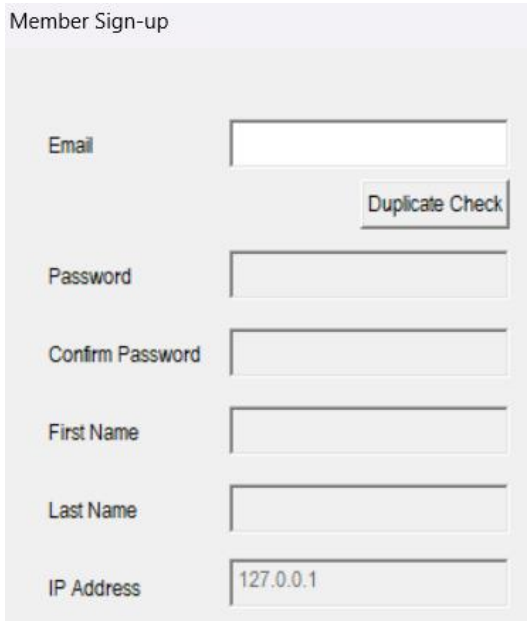
- Use Ethereal site for Fake email service
- Laptops testing the application should be connected through router
- Firewall configuration in Laptop should be disabled

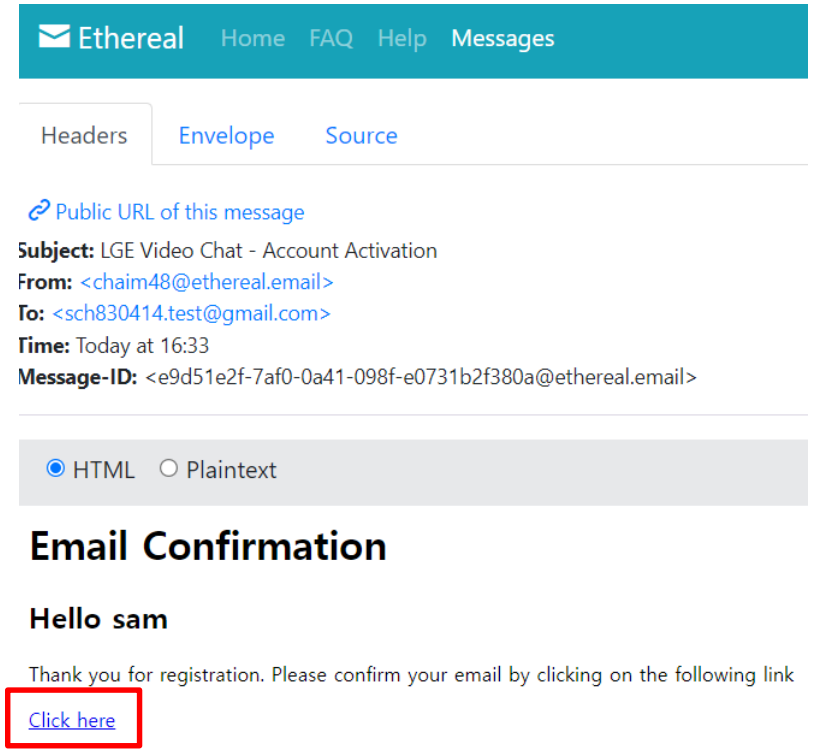
Final test result

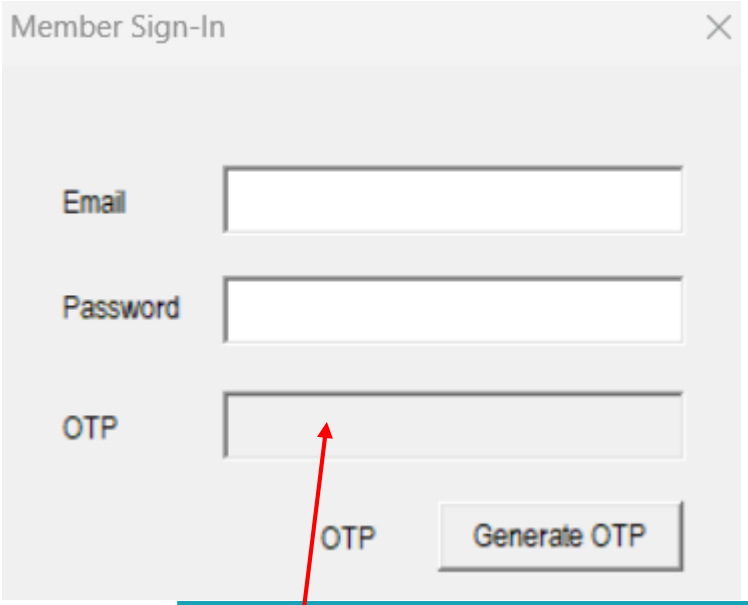
- Total test cases : 47
- Pass : 34, Fail : 13 (not critical issues)
- Pass rate : 72.3%

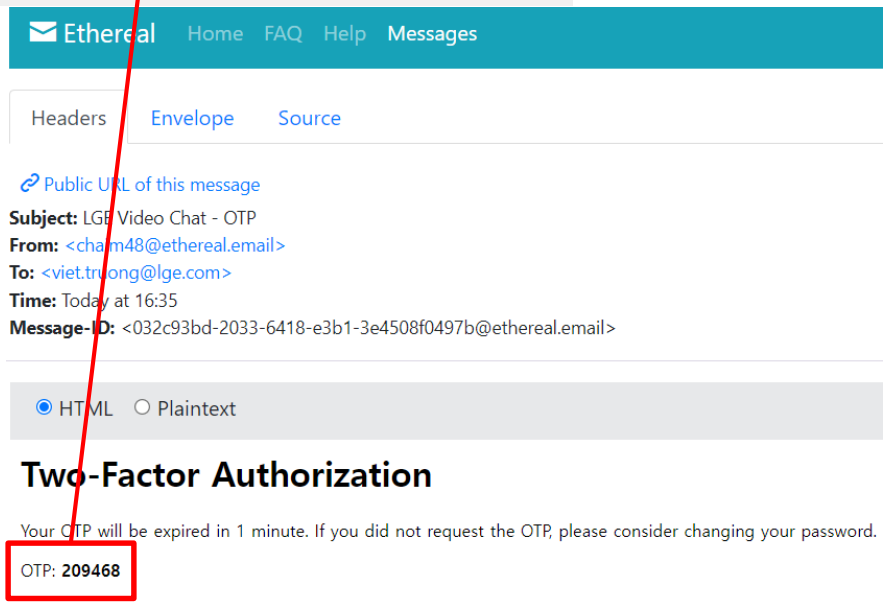
Verification

✓ Two factor authentication





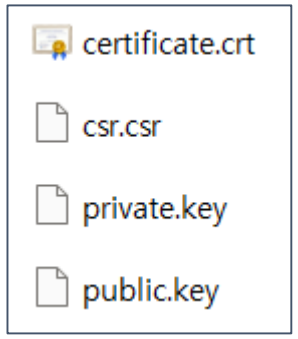




Verification

- ✓ **Server Authentication & Secure communication**

```
192.168.0.212    192.168.0.249    TLSv1.3    347 Client Hello
192.168.0.249    192.168.0.212    TLSv1.3   1555 Server Hello, Change Cipher Spec, Application Data, Applica
192.168.0.212    192.168.0.249    TCP         60 53959 → 10000 [ACK] Seq=294 Ack=1502 Win=262656 Len=0
192.168.0.212    192.168.0.249    TLSv1.3    134 Change Cipher Spec, Application Data
192.168.0.249    192.168.0.212    TLSv1.3    293 Application Data
192.168.0.212    192.168.0.249    TCP         60 53959 → 10000 [ACK] Seq=374 Ack=1741 Win=262400 Len=0
192.168.0.249    192.168.0.212    TLSv1.3    293 Application Data
192.168.0.212    192.168.0.249    TLSv1.3     80 Application Data
```

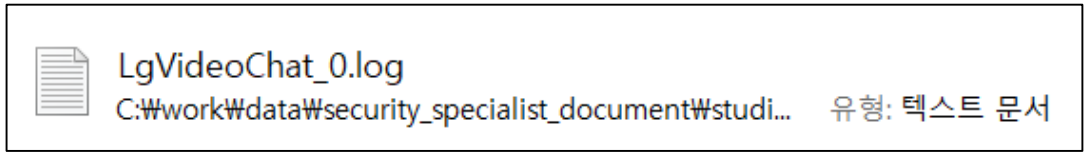


Self signed certificate

```
Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 118
    Version: TLS 1.2 (0x0303)
    Random: be96661c29a05067205bffcfb80a1663ff14eb20b0b10ca95b900fc245bd0e6b
    Session ID Length: 32
    Session ID: f9156460f8e527f7d1515bf36925ae9075a11bf362491d962715580501ac158d
    Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
```

Verification

✓ **Storing log file to the file system**

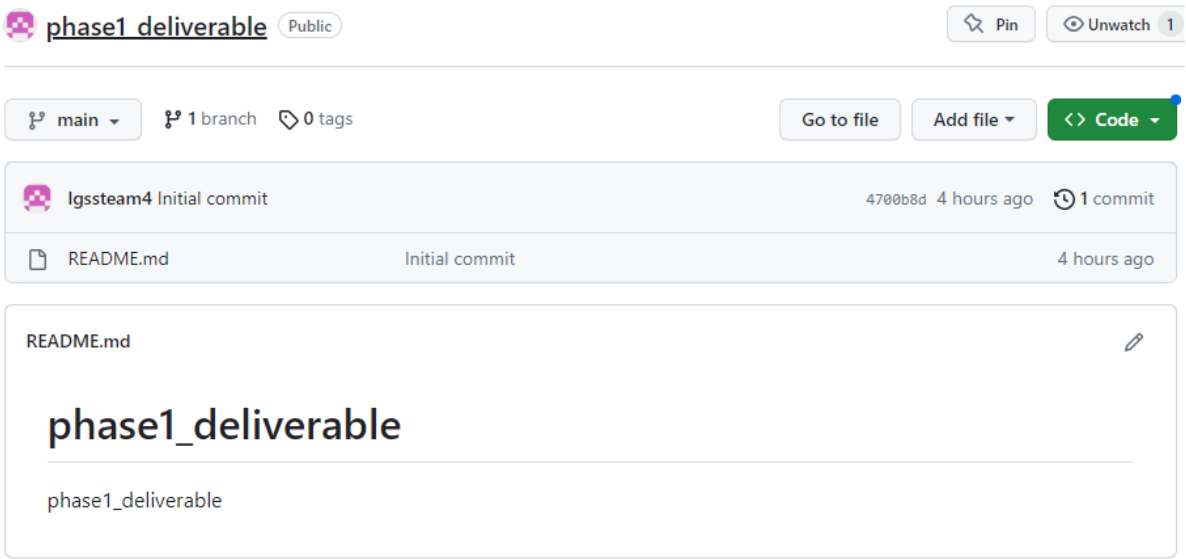



```
2023-06-20, 19:18:30.989024 [00:00:00] <info> Guid = {6C87AC70-38F6-4B01-8A61-5B4C3B65053C}
2023-06-20, 19:27:47.212814 [00:00:00] <info> Guid = {3252D919-9D37-4429-ADE7-D0A73B22A17C}
2023-06-21, 12:07:50.930850 [00:00:00] <info> Guid = {D898B6A0-CCF9-48B7-BB89-723CBE387D16}
2023-06-21, 12:07:55.752581 [00:00:04] <info> RemoteAddress : 192.168.0.249LocalIpAddress : 192.168.0.249
2023-06-21, 12:20:54.122337 [00:00:00] <info> Guid = {7056553E-D2C1-40AC-9778-607ED83DDBD1}
2023-06-21, 12:20:59.333956 [00:00:05] <info> RemoteAddress : 192.168.0.249LocalIpAddress : 192.168.0.249
2023-06-21, 12:22:41.680384 [00:00:00] <info> Guid = {9002495C-FE57-4EE1-A8CF-83FCE881B653}
2023-06-21, 12:22:43.181370 [00:00:01] <info> RemoteAddress : 192.168.0.249LocalIpAddress : 192.168.0.249
2023-06-21, 12:23:33.552701 [00:00:00] <info> Guid = {86421E05-3D75-4EDC-B103-AC7C95F2C335}
2023-06-21, 12:23:35.161014 [00:00:01] <info> RemoteAddress : 192.168.0.249LocalIpAddress : 192.168.0.249
2023-06-21, 12:24:26.536335 [00:00:00] <info> Guid = {BEEE3516-151D-4C78-9E01-E43758C05A7B}
2023-06-21, 12:24:27.894788 [00:00:01] <info> RemoteAddress : 192.168.0.249LocalIpAddress : 192.168.0.249
2023-06-21, 12:26:49.546051 [00:00:00] <info> Guid = {3B1A16BB-5507-4607-9073-23E39BBA2361}
2023-06-21, 12:26:51.329804 [00:00:01] <info> RemoteAddress : 192.168.0.249LocalIpAddress : 192.168.0.249
2023-06-21, 12:27:54.574231 [00:00:00] <info> Guid = {860962F9-68B5-4938-94E8-F46847E46F9F}
2023-06-21, 12:27:56.289588 [00:00:01] <info> RemoteAddress : 192.168.0.249LocalIpAddress : 192.168.0.249
2023-06-22, 17:12:18.078287 [00:00:00] <info> Guid = {9510CFED-F110-44FF-B7F9-7BB528778DCB}
2023-06-22, 17:12:59.253238 [00:00:41] <error> Email is empty
2023-06-22, 17:41:05.521038 [00:28:45] <error> Invalid email format
2023-06-22, 17:41:12.323231 [00:28:52] <error> Password is invalid
2023-06-22, 17:44:16.883501 [00:31:56] <error> rc =0 status_code = 409
2023-06-22, 17:44:16.884501 [00:31:56] <error> Error is occurred, Please check the input value
2023-06-22, 17:44:34.089284 [00:32:13] <error> rc =0 status_code = 409
2023-06-22, 17:44:34.089284 [00:32:13] <error> Error is occurred, Please check the input value
2023-06-22, 17:44:43.727433 [00:32:23] <error> rc =0 status_code = 409
2023-06-22, 17:44:43.727433 [00:32:23] <error> Error is occurred, Please check the input value
2023-06-22, 17:44:48.724972 [00:32:28] <info> User created successfully. Please check your email to activate your account!
2023-06-22, 17:44:48.725971 [00:32:28] <info> User created successfully
2023-06-22, 17:47:23.285006 [00:35:03] <error> Please enter the OTP code that has been sent to your email
2023-06-22, 17:48:35.313869 [00:36:15] <error> You entered the wrong password
If you are wrong more than 2 times, your account will be locked for 1 hour
2023-06-22, 17:48:47.312999 [00:36:27] <error> Please enter the OTP code that has been sent to your email
```

Deliverables

- ✓ **Following materials will be uploaded to Github**
  - Requirement
  - System Design and Sequence diagram
  - Developer guide to build the software
  - User guide to run the software
  - Final source code

- ✓ https://github.com/lgssteam4/phase1_deliverable



Lessons Learned

- ✓ **Project Plan from Security Perspective**
  - Our team could understand overall process for the project which has to consider security
  - To catch up the unexpected needs, our team changed the initial schedule and order
- ✓ **Threat Analysis & Secure Design**
  - Our team was able to realize the importance of threat analysis for secure design
  - Applying only given requirement by customer can be very dangerous from security perspective
  - The more we learn and experience on security, the more we could find the threats and mitigations
- ✓ **Secure Implementation**
  - Using open source libraries were essential for our implementation
  - Not only secure coding but also managing vulnerabilities in the 3rd party libraries will be very important for secure implementation

OSV | Open Source Vulnerabilities

Q&A

Email Contact for Additional Questions and Incident Response : ss-team4@lge.com