Phase 1 Report

Team 3

About This Document

Document Information

Issuing authority	Team 3

Revision History

Verion	Date	Comment	Author	Approver
1.0	2021-06-28	Initial Release	Team 3	

Purpose

This document is a document that summarizes the entire document about the output made during the Phase 1 process.Through this document, you can check what was considered to make AACS in Team3 and how to solve it.

Scope

- Project Plan
- Requirements
- Threat Modeling
- UX
- Architectural Design
- Implementation
- Testing
- Vulnerability Report
- Cryptographic Algorithm
- Remain Works
- Setup Manual

Related Documents

Documents related to this document include :

- Team3_AACS_Requirements
- Team3_AACS_ThreatAnalysis_RiskAssessment_Result
- Team3_AACS_Message_Protocol
- Team3_AACS_Architecture_Design
- Team3_AACS_UX
- Team3_AACS_Testcase
- Team3_AACS_Static_Analysis_Report
- Team3_AACS_Work_Schedule
- Team3_AACS_Opensource_Vulnerability_Report
- Team3_AACS_Remaing_Work
- Team3_AACS_Setup_Manual

Acronyms / Glossary

Acronym	Description
AACS	AI Attendance Check System
FRS	Face Recognition System
ACS	Attendance Check System

Table of Contents

Executive Summary 6
Summary 6
Phase 1 Activity 7
Project Plan 7
Milestone 7
Schedule Plan & Actual Work 7
Requirement Analysis 9
System Scope 9
System Overview 10
Functional Requirements 10
Quality Attribute 14
Threat Modeling 16
Data Flow Diagram 16
Risk Assessment 16
Threat Modeling Summary 22
UX 23
UX Main Page 23
UX Sub Page 23
Architecture Design 25
Architecture Drivers 25
SW Main Features 25
Quality Attribute with Utility Tree 26
Constraints 27
SW Architecture Representations 28
Initial Dynamic View 28
SW Component Descriptions 29
Refined Architecture 30
Dynamic view with security applied after threat analysis 30
Static View 31
Architectural Alternatives 31
Physical View 32
Secure Design Pattern 33
Implementation 34
Protocol Header 34
Protocol Details 34
Static Analysis 35
Static Analysis Summary 35
Testing 37
Test Case 37
Test Case Summary 42
Vulnerability Analysis 42
Vulnerability Analysis Summary 42
Cryptographic Algorithms 43
Secure communication with TLS 1.3 43
Security Manager 43

Cryptographic Algorithms Applied 43
Key Management 44
Crytoperiod 44
References 45
Setup Manual 46
PC Client Setup 46
Server Setup 50
Admin Setup 51
Deployment Setup 52
Remaining works 53
Development Part 53
Attendance Check System(PC Application) 53
Face Recognition System(Jatson nano part) 53
Common System 54
Certificate 54
Jatson Nano Ubuntu 54

Strength / Weakness **55**
Strength 55
Weakness 55

Lesson & Learned **56**
What all members felt 56

1. Executive Summary

1.1. Summary

In phase 1, we experienced the process of applying security to the AI Attendance Check System.By applying the theoretically learned security information to the actual project, I pondered what kind of problems there are and how to solve them.

Threat modeling was applied for additional security in the software process, and security functions were attempted in addition to the basic functions of AACS through security requirements, architecture design for security, and security testing.

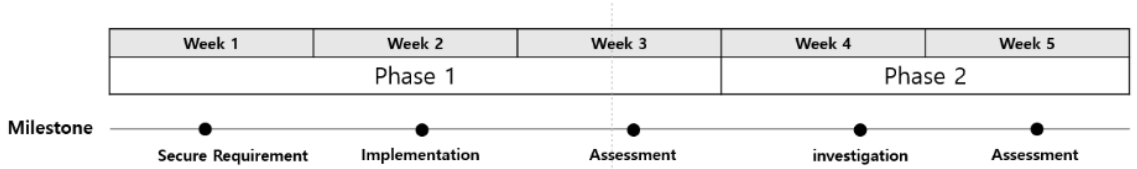
The security design was made according to the security requirements, but all the requirements were not satisfied due to the business constraint that the mission had to be completed within 3 weeks.
All the requirements for the function were derived, but the work that could be done within 3 weeks was selected and implemented.

All members participated in the requirements derivation and design process to understand the project, and in implementation, each member's role was efficiently divided.

2. Phase 1 Activity

2.1. Project Plan

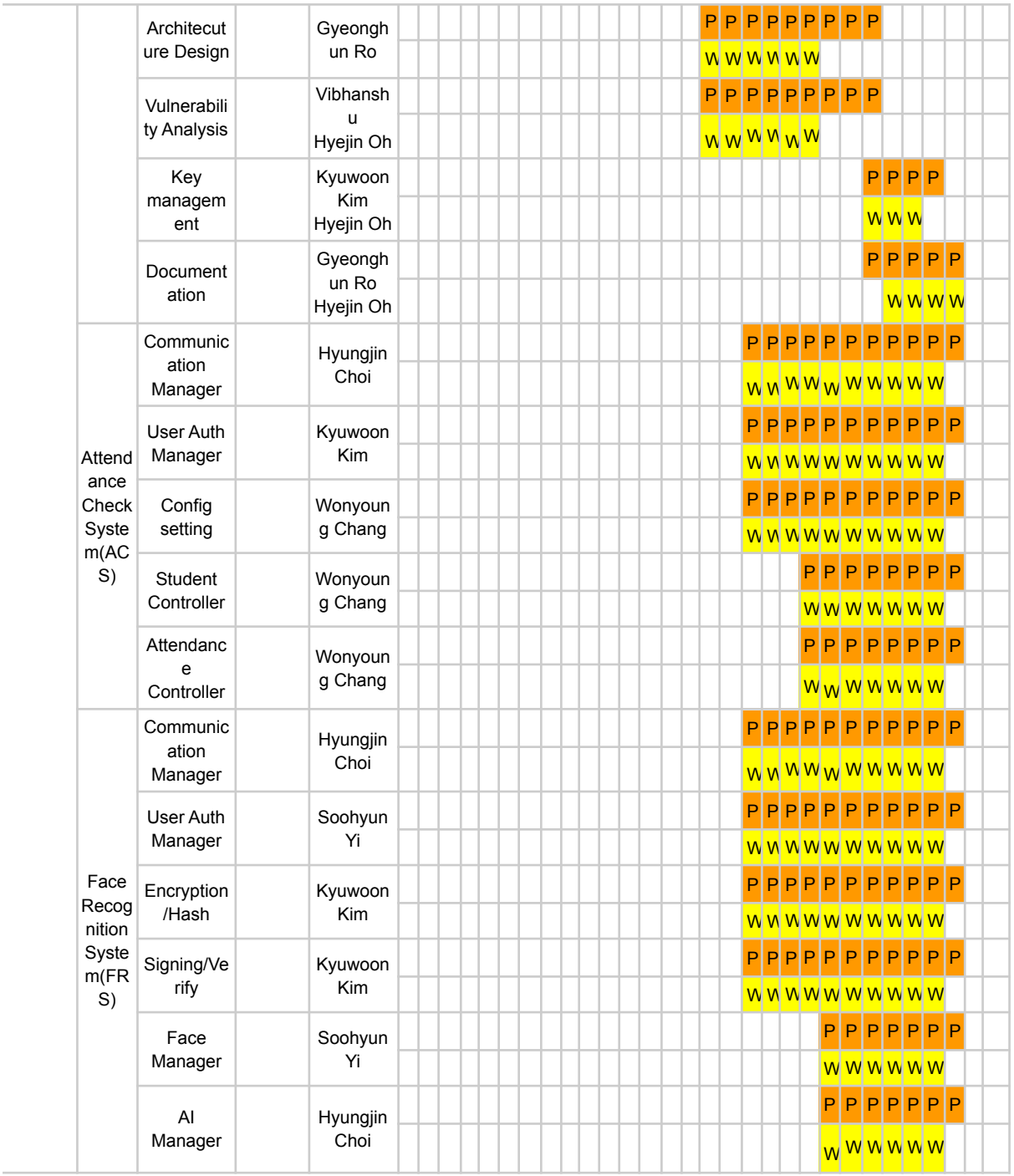
2.1.1. Milestone



Week	Activity	Output	Note
Week 1	Requirement, Threat Modeling, Development Environment, Architecture, Module / Interface design, Vulnerability scanning	requirement specification, architecture, software design	
Week 2	Implementation, Static analysis, key management, crypto algorithm	source code, key management, crypto algorithm	
Week 3	Assessment, Testing	system guide, test case, assessment , testing result, vulnerability report	
Week 4	System Investigation, Vulnerability scanning, Fuzz / Penetrate testing	system investigation	
Week 5	Assessment,	security assessment report	

2.1.2. Schedule Plan & Actual Work

Phase	Part	Level 1	Level 2	Owner	Week 1										Week 2										Week 3									
					1-A	1-P	2-A	2-P	3-A	3-P	4-A	4-P	5-A	5-P	1-A	1-P	2-A	2-P	3-A	3-P	4-A	4-P	5-A	5-P	1-A	1-P	2-A	2-P	3-A	3-P	4-A	4-P	5-A	5-P
Phase 1	Process	Team Building		All	P	P																												
					W	W																												
		Project Plan		Gyeonghun Ro	P	P																												
					W	W																												
		Requirement Analysis		All								P	P	P	P																			
												W	W	W																				
		Project Analysis		All			P	P		P	P	P																						
							W	W		W	W																							
		Risk management		All					P	P																								
									W																									
		Software Design	Data Flow Diagram	All											P	P	P	P																
															W		W	W																
		Threat Modeling		All														P	P	P														
																		W	W	W	W													



*P : Plan, W : Actual Work

2.2. Requirement Analysis

2.2.1. System Scope

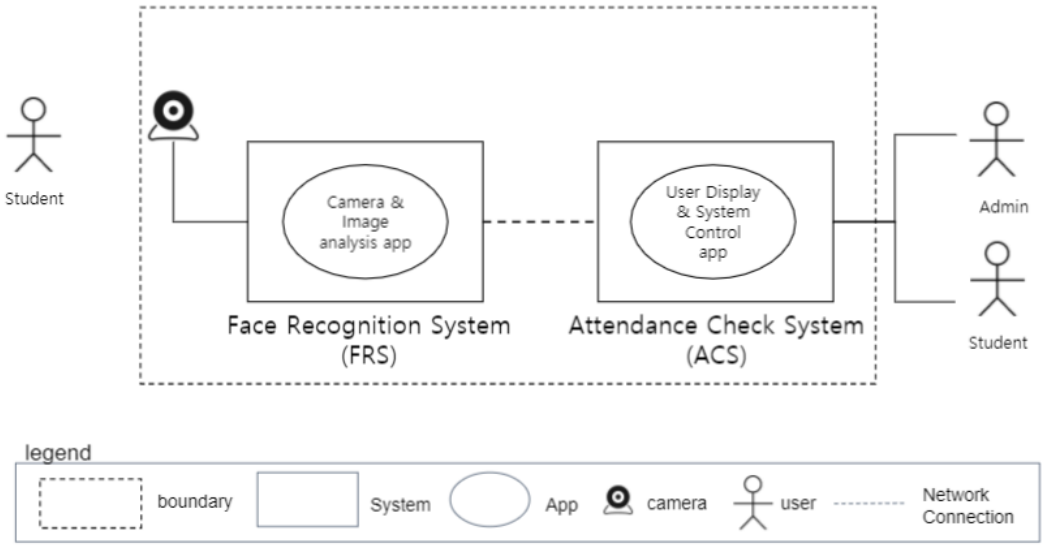
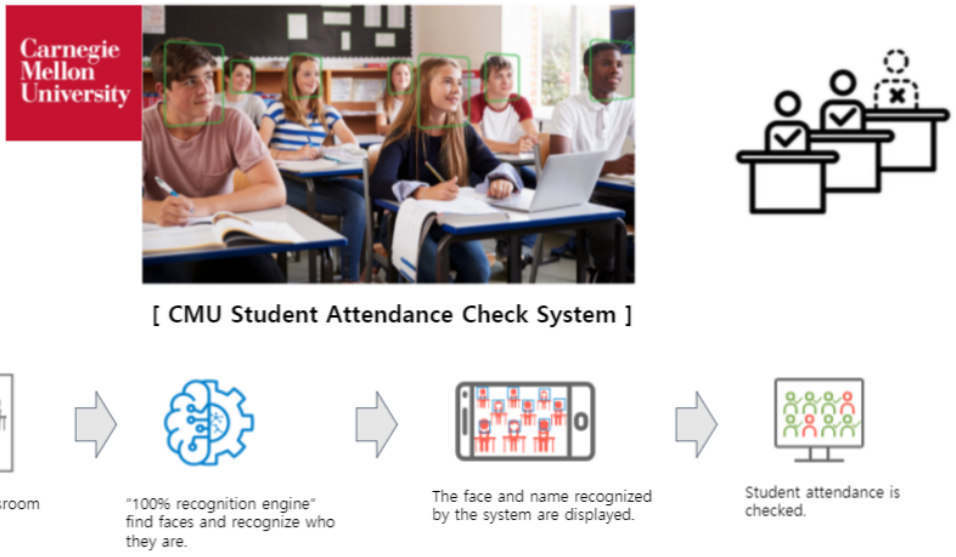


Figure 1 System Scope

2.2.2. System Overview

- The system registers the students' faces through the camera.
- Check the attendance status of registered students according to the set class time. Attendance status is indicated as attendance, tardy, or absence.
- Administrators can check the attendance status of students.
- When a student requests an administrator to check past attendance, it can be viewed as a saved video.



2.2.3. Functional Requirements

Describes system requirements that satisfy customer requirements.

Functional requirements were derived by analyzing customer requirements.

In addition, some requirements were derived through threat analysis.

- ID: Requirement_ID with AACS-REQ-xxx
- category : Categorize and organize requirements. If it is divided into categories, it is thought that it will be helpful to check them by category during future development.
- Contents : Detailed description of requirements
- Priority: Define and describe development priorities. Since our development period is set to 3 weeks, it is necessary to prioritize and apply it across the entire requirements.
- implementation : It is an indication of an implementation application item. Among the total requirements, the requirements reflected in the implementation are indicated.

ID	category - 1	category - 2	contents	priority	type	implementation
AACS-REQ-001	login	create user	This system should be able to create users through membership registration.	Low	common	X
AACS-REQ-00		user password	The system applies a hash function to the user password. SHA256	High	common	O

2						
AACS-REQ-003		user password	To set a strong password we should provide the notice (message) to select Password with combination of chars, numbers and special chars etc	High	common	O
AACS-REQ-004		captcha authentication	This system makes a captcha appear when the password is wrong 3 times or more.	Low	common	X
AACS-REQ-005		user authentication	This system requires 2-factor authentication for user authentication.	High	common	O
AACS-REQ-006		set permissions	This system should be able to set student/administrator privileges when creating users.	High	common	O
AACS-REQ-007		set mode	This system should be able to set secure mode and real-time mode when logging in.	High	common	O
AACS-REQ-008		login	This system must be able to connect the ACS and FRS systems during the login function.	High	common	O
AACS-REQ-009		pre-defined user	This system creates users with the privileges of students and administrators in advance.	Mid	common	O
AACS-REQ-010	register of student (Learn mode)	add photo	The system should be able to add photos of students.	High	common	O
AACS-REQ-011		delete photo	The system should be able to delete student photos.	High	derived	O
AACS-REQ-012		end register	The system must be able to finish the student registration function.	High	derived	O
AACS-REQ-013		tune photo	This system should provide the ability to adjust the brightness and illuminance of the photo.	Low	common	X
AACS-REQ-014		select class	This system should allow students to set up their own classes.	Low	derived	X
AACS-REQ-015	attendance check (Run mode, Test run mode)	live attendance	The system should be able to check student attendance in real time.	High	common	O
AACS-REQ-016		past attendance	This system should be able to check the attendance of students through past saved images.	High	common	O
AACS-REQ-01		student list	The system should be able to see a list of students assigned to a class.	High	derived	O

7						
AACS-REQ-018		save student attendance time	The system should be able to store the attendance time of students.	High	derived	O
AACS-REQ-019		check student status	The system should be able to know the attendance, late, and absence status of students.	High	derived	O
AACS-REQ-020		Attendance time setting	The system should be able to set attendance times for students.	High	derived	O
AACS-REQ-021		add class	This system should be able to add a class for attendance check.	Low	derived	X
AACS-REQ-022		Add class manager	This system should be able to set the person responsible for the class.	Low	derived	X
AACS-REQ-023	security	log	The system must support logging for non-repudiation.	High	common	X
AACS-REQ-024		defense	The system should be prepared for keyboard and mouse hooking.	High	common	X
AACS-REQ-025		storing personal information	In this system, the user's personal information and data related to facial recognition must be encrypted and stored.	High	common	O
AACS-REQ-026		secure mode	This system requires encryption of data transmission.	High	common	O
AACS-REQ-027		secure code	Secure coding & static analysis -> Fix RATS results	High	common	O (partially)
AACS-REQ-028		snooping	The system should not allow intermediaries to snoop or spy on the ongoing video feed.	High	common	O
		The security requirements are derived through threat analysis as follows.				
AACS-REQ-029		limited photos	Each student must be able to save up to 5 photos.	High	derived	O
AACS-REQ-030		check capacity	When saving student photos, this system should be able to check the remaining capacity.	High	derived	X
AACS-REQ-031		check capacity	When saving a video about attendance, this system should be able to check the remaining capacity.	Midium	derived	X
AACS-		separate	This system should be able to save the video	Midium	com	X

REQ-032		partition	of the attendance in a separate partition.		mon	
AACS-REQ-033		operation	This system should be able to operate the attendance function even if it is not possible to save the video of the attendance.	High	common	X
AACS-REQ-034		encryption	Videos of attendance in this system must be encrypted.	High	common	X
AACS-REQ-035		hash	User accounts accessing this system must be hashed.	High	common	O
AACS-REQ-036		hash	Config setting file that manages device information stored in the system should be signed	High	derived	O
AACS-REQ-037		encryption	Face DB in the system must be encrypted.	High	common	X
AACS-REQ-038		input validation	When logging into the system, the input data should be verified.	High	common	X
AACS-REQ-039		sign	Face DB data stored in the system must be signed by admin.	High	common	X
AACS-REQ-040		encryption	Video DB data stored in the system must be encrypted.	High	common	X
AACS-REQ-041		sign	Video DB data stored in the system must be signed by admin.	High	common	X
AACS-REQ-042		encryption	User DB data stored in the system must be encrypted.	High	common	O
AACS-REQ-043		sign	User DB data stored in the system must be signed by admin.	High	common	O
AACS-REQ-044		heart beat	The system's Comm Manager (ACS) must apply a heart beat.	High	common	X
AACS-REQ-045		input validation	Input verification for the Config setting in the system should be done.	High	common	O
AACS-REQ-046		hash	AI Engine data shall be hashed.	High	common	O
AACS-REQ-04		input validation	Input verification for engine data in the system should be done.	High	common	O

7						
AACS-REQ-048		data loading	It is necessary to check the loading completion of the engine data of the system.	High	derived	O
AACS-REQ-049		input validation	The Comm Manager (FRS) in the system should verify the input.	High	common	O
AACS-REQ-050		IP filtering	The FRS system must receive only a set IP through IP filtering.	High	common	X
AACS-REQ-051		TLS	TLS version 1.2 and above is needed must be applied for communication between FRS and ACS in the system.	High	common	O

legend

type : common-initial requirements, derived-AACS requirements

implementation : X-won't do, O-will do

2.2.4. Quality Attribute

Quality attribute refers to the characteristic attributes of a product. Satisfying quality attributes can satisfy customers' requirements for quality.

Quality attributes were derived through customer requirements.

In addition, some Quality attributes were derived through threat analysis.

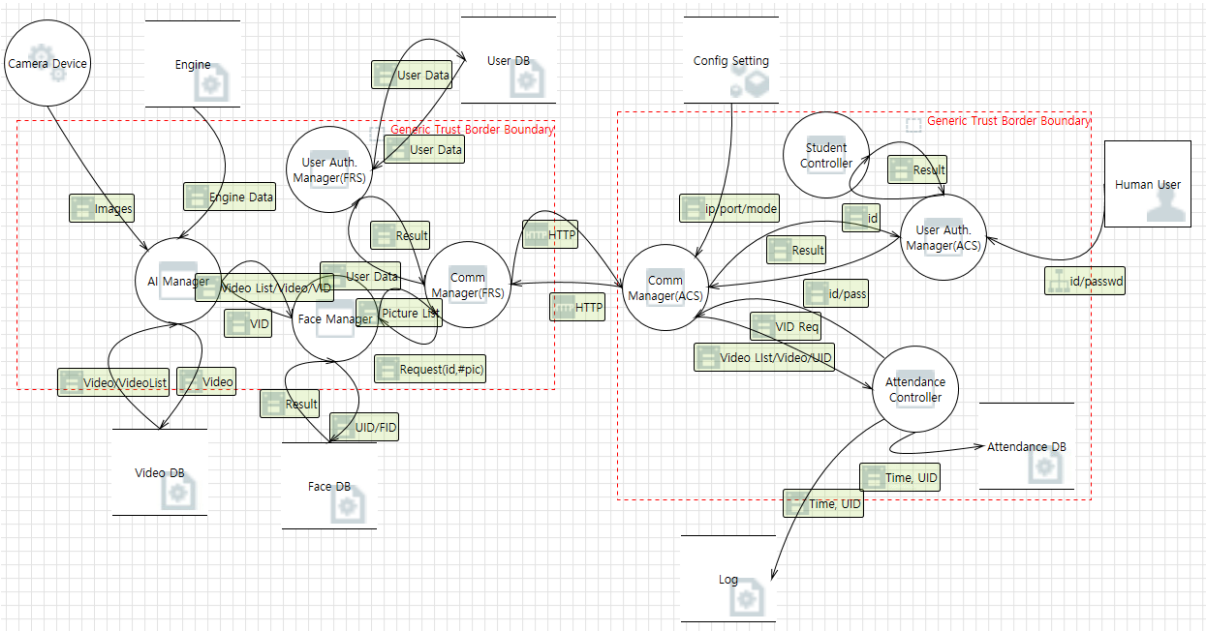
1. ID: Quality Attribute ID with AACS-QA-xxx
2. Properties : Types of quality attributes
3. Contents : Detailed description of quality attribute

ID	Properties	Contents
AACS-QA-001	Performance	The system must deliver video as close to real time as possible, especially in real-time mode.
AACS-QA-002	Authentication	The system must use two factor authentication for sign on and user credentials must be protected. Lost or compromised credentials must be handled in a reasonable way.
AACS-QA-003	Communication privacy	When in the desired mode the system must ensure that data sent to a user remains private while in transit. No intermediary should be able to snoop or spy on an ongoing video feed.
AACS-QA-004	Proof of identity	Users should be confident that the camera they are using is the one that they believe it is.

	(nonrepudiation)	
AACS-QA-005	Multi-user privacy:	The system must ensure that multiple video feeds remain private between the intended users.
AACS-QA-006	reliability	The system must ensure that video is reliably delivered. The system should recover from networking errors as soon as possible. The goal is to maintain a secure, performant connection at all costs.
AACS-QA-007	Testing	Ensure the developed software is adequately tested.
AACS-QA-008	Availability	Conduct proper fault/error detection, recovery and reporting.
AACS-QA-009	Security	Ensure the developed software adheres to the company coding standard and quality standards.
AACS-QA-010	Security	Key management for system must be secure.

2.3. Threat Modeling

2.3.1. Data Flow Diagram



2.3.2. Risk Assessment

ID	TM-ID	Title	Category	Interaction	Description(Detail of Security Threats)	Justification	**Likelyhood**	**Impact**	**Risk Assessment Result**	Mitigation	**Security Requirement**
1	4	Potential Excessive Resource Consumption for Face Manager or Face DB	Denial Of Service	UID /FID	Does Face Manager or Face DB take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.	Face DB consumes disk resource. If a number of pictures has no limit, resource consumption attacks are possible.	High	High	Critical	1. Limit the number of pictures per user 2. The remaining storage check shall be applied,	1. Each student must be able to save up to 5 photos. 2. When saving student photos, this system should be able to check the remaining capacity.
2	6	Potential Excessive Resource Consumption for AI Manager or Video DB	Denial Of Service	Video	Does AI Manager or Video DB take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are	Video DB consumes disk resource. If time of recoding has no limit, resource consumption attacks are	High	High	Critical	The remaining storage check shall be applied,	1. When saving a video about attendance, this system should be able to check the remaining capacity. 2. This system should be able to

					times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.	possible.					save the video of the attendance in a separate partition. 3. This system should be able to operate the attendance function even if it is not possible to save the video of the attendance.
3	8	Weak Access Control for a Resource	Information Disclosure	Video/VideoList	Improper data protection of Video DB can allow an attacker to read information not intended for disclosure. Review authorization settings.	Video should be not opened.	Medium	High	High	Video DB shall be encrypted	1. Videos of attendance in this system must be encrypted.
4	16	Spoofing of Source Data Store User DB	Spoofing	User Data	User DB may be spoofed by an attacker and this may lead to incorrect data delivered to User Auth. Manager(FRS). Consider using a standard authentication mechanism to identify the source data store.	User DB is file and if attacker change it to link, then User DB may be spoofed and Auth Manager may take incorrect id and password, then attacker can be log-in successfully.	Medium	High	High	User DB shall be encrypted	1. User accounts accessing this system must be hashed.
5	17	Weak Access Control for a Resource	Information Disclosure	User Data	Improper data protection of User DB can allow an attacker to read information not intended for disclosure. Review authorization settings.	User Data should not be opened. It has user id and password.	Medium	High	High	User Data (login id, password) shall be encrypted	1. User accounts accessing this system must be hashed.
6	21	Spoofing of Source Data Store Config Setting	Spoofing	ip/port/mode	Config Setting may be spoofed by an attacker and this may lead to incorrect data delivered to Comm Manager(ACS). Consider using a standard authentication mechanism to identify the source data store.	Config Setting is file and if attacker change it to link, then Config Setting may be spoofed and Comm Manager(ACS) may take incorrect connection configuration, then Comm Manager(ACS) may connect to attacker's FRS.	High	High	Critical	Configuration setting file shall be signed to prevent tempering attack.	1. Config setting file that manages device information stored in the system should be hashed
7	65	Spoofing the Face Manager Process	Spoofing	UID/FID	Face Manager may be spoofed by an attacker and this may lead to unauthorized access	FaceDB is one of the important assets. Need	High	High	Critical	Face DB shall be encrypted	1.Face DB in the system must be encrypted.

					to Face DB. Consider using a standard authentication mechanism to identify the source process.	to consider authentication mechanism,					
8	56	Potential Lack of Input Validation for User Auth. Manager(ACS)	Tampering	id/passwd	Data flowing across id/passwd may be tampered with by an attacker. This may lead to a denial of service attack against User Auth. Manager(ACS) or an elevation of privilege attack against User Auth. Manager(ACS) or an information disclosure by User Auth. Manager(ACS). Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.	Window login input UI can be the important attack entry to attacker. Input data validation is neede. Need to integrate the attack paths below later.	High	High	Critical	1. Input validation for login information is needed. '- minimum length, maximum length - not allow command - Password complexity (Upper Letter, Lower Letter, Special Character) 2. keylogger prevention system can be applied.	1. When logging into the system, the input data should be verified.
9	66	The Face DB Data Store Could Be Corrupted	Tampering	UID/FID	Data flowing across UID/FID may be tampered with by an attacker. This may lead to corruption of Face DB. Ensure the integrity of the data flow to the data store.	Data stored in Face DB must have integrity.	Medium	High	High	Face DB shall be signed to prevent tempering attack.	1. Face DB data stored in the system must be signed by admin.
10	68	Data Flow Sniffing	Information Disclosure	UID/FID	Data flowing across UID/FID may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.	User image exposure must be prevented.	Medium	High	High	User images Face DB shall be encrypted	1. Face DB data stored in the system must be encrypted.
11	71	Spoofing the AI Manager Process	Spoofing	Video	AI Manager may be spoofed by an attacker and this may lead to unauthorized access to Video DB. Consider using a standard	Access to Video DB should be strictly prohibited.	High	High	Critical	Video DB shall be encrypted	1. Video DB data stored in the system must be encrypted.

					authentication mechanism to identify the source process.						
12	72	The Video DB Data Store Could Be Corrupted	Tampering	Video	Data flowing across Video may be tampered with by an attacker. This may lead to corruption of Video DB. Ensure the integrity of the data flow to the data store.	The integrity of the video DB data must be guaranteed.	Medium	High	High	Video DB shall be signed to prevent tempering attack.	1. Video DB data stored in the system must be signed by admin.
13	74	Data Flow Sniffing	Information Disclosure	Video	Data flowing across Video may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.	Video DB data should be managed so that it is not exposed.	Medium	High	High	Video DB shall be encrypted	1. Video DB data stored in the system must be encrypted.
14	104	Spoofing the User Auth. Manager(FRS) Process	Spoofing	User Data	User Auth. Manager(FRS) may be spoofed by an attacker and this may lead to unauthorized access to User DB. Consider using a standard authentication mechanism to identify the source process.	It should be treated with high priority	High	High	Critical	User DB shall be encrypted	1. User DB data stored in the system must be encrypted.
15	105	The User DB Data Store Could Be Corrupted	Tampering	User Data	Data flowing across User Data may be tampered with by an attacker. This may lead to corruption of User DB. Ensure the integrity of the data flow to the data store.	User DB data should be protected for tampering	Medium	High	High	User DB shall be signed to prevent tempering attack.	1. User DB data stored in the system must be signed by admin.
16	107	Data Flow Sniffing	Information Disclosure	User Data	Data flowing across User Data may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.	User DB data should be protected for sniffing attacks	Medium	High	High	User DB shall be encrypted	1. User DB data stored in the system must be encrypted.

17	110	Spoofing the User Auth. Manager(FRS) Process	Spoofing	User Data	User Auth. Manager(FRS) may be spoofed by an attacker and this may lead to information disclosure by User DB. Consider using a standard authentication mechanism to identify the destination process.	It should be treated with high priority	High	High	Critical	User DB shall be encrypted	1. User DB data stored in the system must be encrypted.
18	130	Spoofing the Comm Manager(ACS) Process	Spoofing	ip/port/mode	Comm Manager(ACS) may be spoofed by an attacker and this may lead to information disclosure by Config Setting. Consider using a standard authentication mechanism to identify the destination process.	Configuration information should be protected for spoofing attacks	High	High	Critical	Configuration setting file shall be signed to prevent tempering attack.	1. Config setting file that manages device information stored in the system should be hashed
19	132	Potential Process Crash or Stop for Comm Manager(ACS)	Denial Of Service	ip/port/mode	Comm Manager(ACS) crashes, halts, stops or runs slowly; in all cases violating an availability metric.	Comm Manager (ACS) should be available for service.	Medium	High	High	Process health check is needed.	1. The system's Comm Manager (ACS) must apply a heart beat.
20	136	Elevation by Changing the Execution Flow in Comm Manager(ACS)	Elevation Of Privilege	ip/port/mode	An attacker may pass data into Comm Manager(ACS) in order to change the flow of program execution within Comm Manager(ACS) to the attacker's choosing.	Comm Manager (ACS) needs to be prepared for this attack.	Medium	High	High	Input validation is needed for Config setting.	1. Input verification for the Config setting in the system should be done.
21	138	Spoofing of Source Data Store Engine	Spoofing	Engine Data	Engine may be spoofed by an attacker and this may lead to incorrect data delivered to AI Manager. Consider using a standard authentication mechanism to identify the source data store.	AI Manager needs to be prepared for this attack.	High	High	Critical	AI Engine data shall be signed to prevent tempering attack.	1. AI Engine data shall be hashed.
22	141	Potential Process Crash or Stop for AI Manager	Denial Of Service	Engine Data	AI Manager crashes, halts, stops or runs slowly; in all cases violating an availability metric.	AI Manager should be available for service.	Medium	High	High	Input validation is needed for AI Engine Input. Process health check is needed.	1. Input verification for engine data in the system should be done. 2. It is necessary to check the loading completion of the engine data of the system.

23	165	Potential Process Crash or Stop for Comm Manager(FRS)	Denial Of Service	HTTP	Comm Manager(FRS) crashes, halts, stops or runs slowly; in all cases violating an availability metric.	Comm Manager(FRS) should be available for service.	High	High	Critical	Input validation for login information in Window App is needed. IP filtering is need to consider.	1. The Comm Manager (FRS) in the system should verify the input. 2. The FRS system must receive only a set IP through IP filtering.
24	164	Data Flow Sniffing	Information Disclosure	HTTP	Data flowing across HTTP may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.	Http is vulnerable to this attack. This requires preparation.	High	High	Critical	To apply TLS version 1.2 and above is needed,	1. TLS version 1.2 and above is needed must be applied for communication between FRS and ACS in the system.
25	162	Potential Lack of Input Validation for Comm Manager(FRS)	Tampering	HTTP	Data flowing across HTTP may be tampered with by an attacker. This may lead to a denial of service attack against Comm Manager(FRS) or an elevation of privilege attack against Comm Manager(FRS) or an information disclosure by Comm Manager(FRS). Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.	Http is vulnerable to this attack. This requires preparation.	High	High	Critical	To apply TLS version 1.2 and above is needed,	1. TLS version 1.2 and above is needed must be applied for communication between FRS and ACS in the system.
26	161	Spoofing the Comm Manager(FRS) Process	Spoofing	HTTP	Comm Manager(FRS) may be spoofed by an attacker and this may lead to information disclosure by Comm Manager(ACS). Consider using a standard authentication mechanism to identify the destination process.	Communication between Comm Manager and Comm Manager (ACS) should be secure.	High	High	Critical	Mutual authentication is needed.	1. Mutual authentication must be performed between FRS and ACS in the system.

27	160	Spoofing the Comm Manager(ACS) Process	Spoofing	HTTP	Comm Manager(ACS) may be spoofed by an attacker and this may lead to unauthorized access to Comm Manager(FRS). Consider using a standard authentication mechanism to identify the source process.	Communication between Comm Manager and Comm Manager (ACS) should be secure.	High	High	Critical	Mutual authentication is needed.	1. Mutual authentication must be performed between FRS and ACS in the system.
28	174	Data Flow Sniffing	Information Disclosure	HTTP	Data flowing across HTTP may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.	FRS->ACS packet sniff(sniff)	High	High	Critical	To apply TLS version 1.2 and above is needed,	1. TLS version 1.2 and above is needed must be applied for communication between FRS and ACS in the system.
29	190	Spoofing the Camera Device Process	Spoofing	Images	Camera Device may be spoofed by an attacker and this may lead to unauthorized access to AI Manager. Consider using a standard authentication mechanism to identify the source process.	Camera -> Ai manager(spoof)	Medium	High	High	Need to consider the way how camera can not be detached physically.	1. Reinforce the hardware to prevent the camera cable from being disconnected

2.3.3. Threat Modeling Summary

Threat Analysis	Risk assessment for high priority threats	Security Requirements from threat mitigation
138	**29**	**23**

The Threat Modeling process was the first process I tried, so I was a little clumsy, but I tried to proceed according to the process.

Through the risk assessment process, we were able to derive 23 security requirements by drawing DFD through the MS Threat Modeling tool and analyzing 140 threats derived based on STRIDE.

2.4. UX

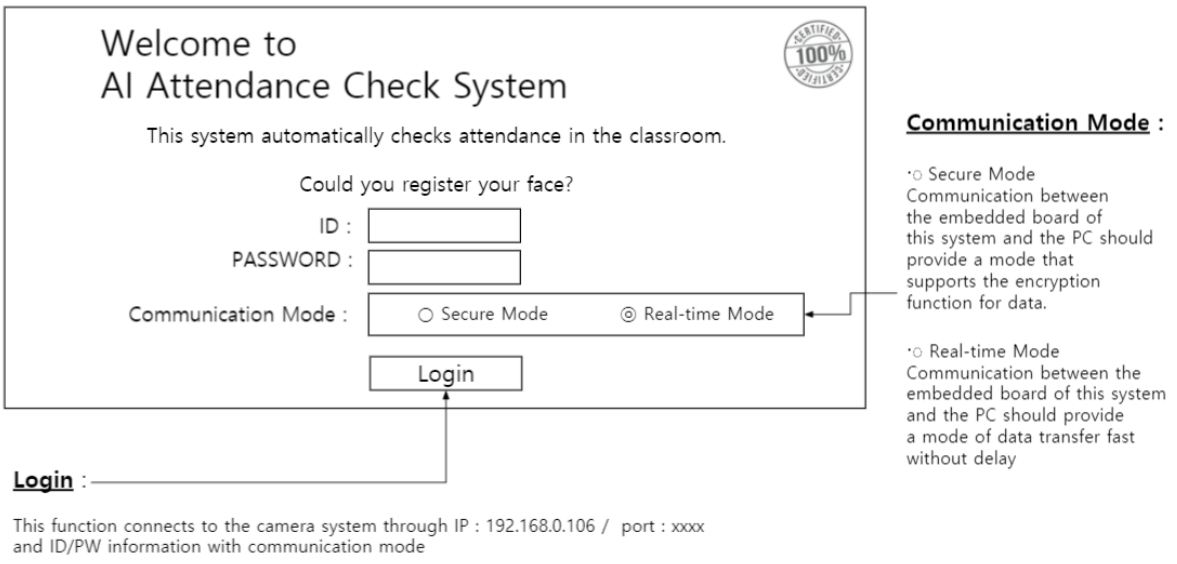
2.4.1. UX Main Page

AI Attendance Check System

Enjoy a cup of coffee during the attendance check time

2.4.2. UX Sub Page

Chapter	Chapter 1	ID	AACS-UX-01	Note	
Location	Initial Page	Side	PC		



Chapter		ID		Note	
Location	Student login	Side	PC		

Add Picture

This function is designed to register the face of a registered student. Students can register up to 5 photos.

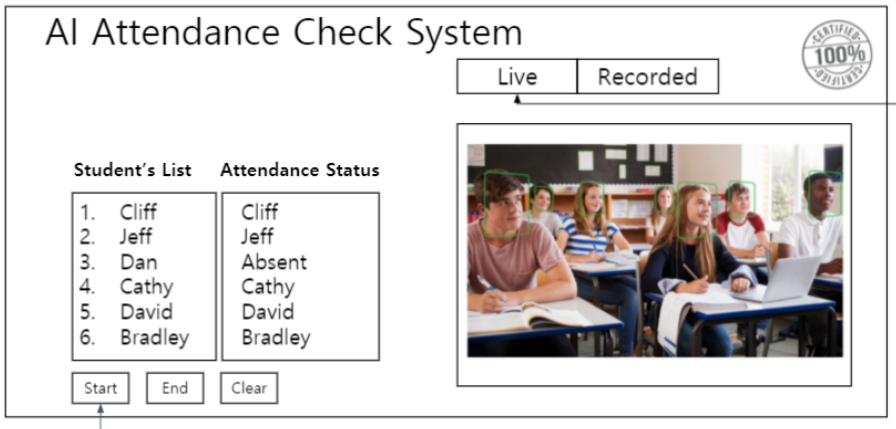
Delete Picture

This is a function to delete a registered student's face photo.

Finish

This function closes the menu for students.

Chapter		ID		Note	
Location	Admin login	Side	PC		



Live :

This is a mode to check attendance live. Attendance, lateness, and absence will be determined based on attendance time.

Recorded :

This is a mode to check attendance through videos saved in the past. Attendance, lateness, and absence will be determined based on attendance time.

Start

This is the menu to start attendance mode on Live. When you start attendance mode, it is automatically saved as a video. Later, the administrator can check attendance through this video.

End

This is the menu to end attendance mode on Live. After exiting the attendance mode, the video recording is automatically stopped.

Clear

If you select this menu, the displayed student attendance time is cleared.

2.5. Architecture Design

2.5.1. Architecture Drivers

2.5.1.1. SW Main Features

Table 1 Software Main Features 1

Level 1	Level 2	Level 3	Descriptions
AACS	FRS	Communication	-Communication on the system is between FRS and ACS. -Provides a function for secure communication between FRS and ACS. -A protocol is defined for communication between two systems, and communication is performed accordingly.
		Face Recognition	-Provides a face recognition function to check the attendance of students in the class. -Face recognition provides the process of registering new students and the ability to recognize registered students through images.
		User Auth	-Provides authentication function for users. -The user information transmitted from ACS is compared with the authorized user information in the system for authentication.
	ACS	Communication	-Communication on the system is between FRS and ACS. -Provides a function for secure communication between FRS and ACS. -A protocol is defined for communication between two systems, and communication is performed accordingly.
		User Auth	-Provides authentication function for users. -The information received from the user through the UI is transmitted to the FRS system to check whether the user is a valid user.
		UI	-Provides a UI for system users. -When a student accesses the system, it provides a function to register his or her face. -When the administrator accesses the system, he or she can view the attendance status of the students. You can check the current attendance status and past attendance status through saved videos.

2.5.1.2. Quality Attribute with Utility Tree

Quality attribute refers to the characteristic attributes of a product. Satisfying quality attributes can satisfy customers' requirements for quality.

Quality attributes were derived through customer requirements.

In addition, some Quality attributes were derived through threat analysis.

1. ID: Quality Attribute ID with AACS-QA-xxx
2. Properties : Types of quality attributes
3. Contents : Detailed description of quality attribute
4. Importance : Scoring importance to the system on a scale of 1 to 5.
5. Difficulty : The difficulty in implementing the Quality Attribute is scored on a scale of 1 to 5.
6. Priority : Importance x Difficulty = Priority given by Quality Attribute. Accordingly, the quality attribute that this system should have is determined.

ID	Properties	Contents	Importance	Difficulty	Priority
AACS-QA-001	Performance	The system must deliver video as close to real time as possible, especially in real-time mode.	2.5	4	10
AACS-QA-002	Authentication	The system must use two factor authentication for sign on and user credentials must be protected. Lost or compromised credentials must be handled in a reasonable way.	4	3	12
AACS-QA-003	Communication privacy	When in the desired mode the system must ensure that data sent to a user remains private while in transit. No intermediary should be able to snoop or spy on an ongoing video feed.	5	4	**20**
AACS-QA-004	Proof of identity (nonrepudiation)	Users should be confident that the camera they are using is the one that they believe it is.	2	4	8
AACS-QA-005	Multi-user privacy:	The system must ensure that multiple video feeds remain private between the intended users.	4	3	12
AACS-QA-006	reliability	The system must ensure that video is reliably delivered. The system should recover from networking errors as soon as possible. The goal is to maintain a secure, performant connection at all	2.5	4.5	11.25

		costs.			
AACS-QA-007	Testing	Ensure the developed software is adequately tested.	3	3.5	10.5
AACS-QA-008	Availability	Conduct proper fault/error detection, recovery and reporting.	4	4	16
AACS-QA-009	Security	Ensure the developed software adheres to the company coding standard and quality standards.	4	4	16
AACS-QA-010	Security	Key management for system must be secure.	4	4.5	18

As above, the main Quality Attribute that the system should have is internally determined for items 3 and 10 according to the priority according to Importance and Difficulty. We will think about the design direction to achieve this.

2.5.1.3. Constraints

Describe the restrictions on this system.

List the business and technical limitations of this system.

This constraint will serve as a driver for architectural design.

ID	constraints	Summary	Contents
AACS-Const-001	Business constraints	Development schedule	Phase 1 period is 3 weeks
AACS-Const-002		Budget issue	No additional budget for development environment
AACS-Const-003	Technical constraints	Development Language	C/C++
AACS-Const-004		Development Board (Jatson Nano)	GPU: 128-core NVIDIA Maxwell™ architecture-based GPU CPU: Quad-core ARM® A57 Video: 4K @ 30 fps (H.264/H.265) / 4K @ 60 fps (H.264/H.265) encode and decode Camera: MIPI CSI-2 DPHY lanes, 12x (Module) and 1x (Developer Kit) Memory: 4 GB 64-bit LPDDR4; 25.6 gigabytes/second Connectivity: Gigabit Ethernet OS Support: Linux for Tegra® Module Size: 70mm x 45mm Developer Kit Size: 100mm x 80mm
AACS-Const-005		PC Development Tool	MS Visual Studio
AACS-Const-006		Router	TP-Link ac1750

AACS-Const -007		no physical modifications	no physical modifications to Jetson Nano.
AACS-Const -008		selected router	use of the supplied and configured router.

2.5.2. SW Architecture Representations

2.5.2.1. Initial Dynamic View

FRS consists of a component for performing a function for recognizing faces to enroll students.
ACS is composed of components to provide a function for attendance to students and administrators by providing UI.
A component is defined between the ACS and FRS system, and data flowing through the component is defined.

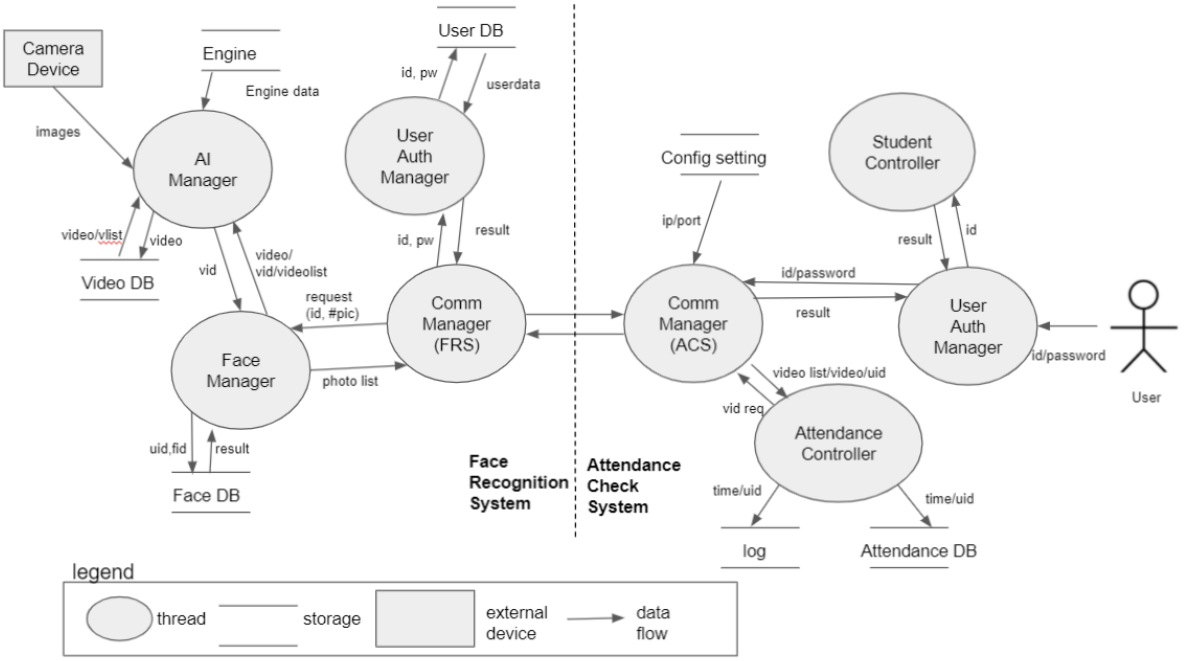


Figure 2 Dynamic View

2.5.2.2. SW Component Descriptions

Below table describes the software components which are specified in the chapter 2.1.

Table 2 SW Component Descriptions

System	Component	R&R	Description
FRS	AI Manager	This component provides the ability to recognize a person's face when attending.	This component provides the ability to verify faces through the AI engine.
	Face Manager	This component manages face photos registered as students.	This component manages student faces through face DB.
	Comm Manager	This component provides functions for communication between FRS and ACS.	Communication function provides secure mode in addition to real-time mode.
	User Auth Manager	This component provides functions for user authentication.	For user authentication, it is checked whether the user is a previously registered user through the user DB.
ACS	Attendance Controller	This component provides a function to show the students in attendance through the UI.	This component provides a function to check student attendance, tardiness, and absence based on school attendance time.
	Student Controller	This component provides a UI function for registering a student's face.	This component provides the ability to add and delete student photos.
	Comm Manager	This component provides functions for communication between FRS and ACS.	Communication function provides secure mode in addition to real-time mode.
	User Auth Manager	This component provides functions for user authentication.	Provides a function to receive user information from the UI for user authentication.

2.5.3. Refined Architecture

2.5.3.1. Dynamic view with security applied after threat analysis

The design below is an architecture that reflects the security requirements derived through risk assessment based on the contents analyzed through threat analysis.

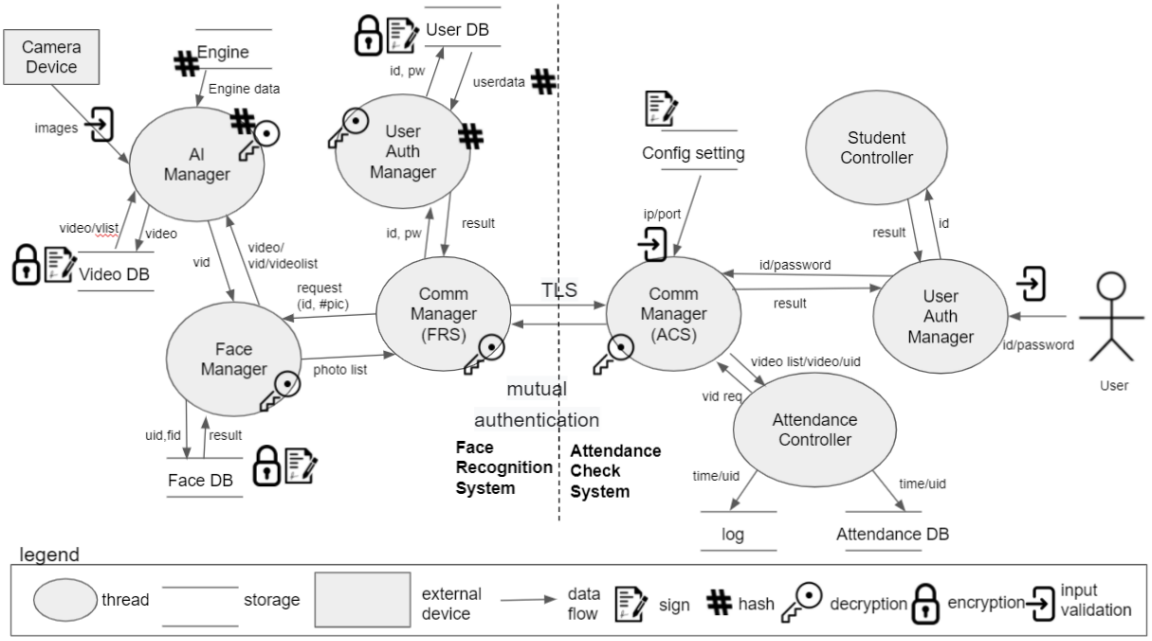


Figure 2 Dynamic View

The above architecture reflects the security requirement and the following functions are added.

1. Encryption and sign are applied to the DB managed by the system to correspond to confidentiality and integrity.
2. Input validation was applied to prevent errors in the values input through the system.
3. The TLS method is applied to secure communication between systems.
4. Hash is applied for security such as password in the system.

2.5.3.2. Static View

Through the AACS static view, it could show the modules for each layer of the system.
A security layer has been added to satisfy the security requirement as shown below.

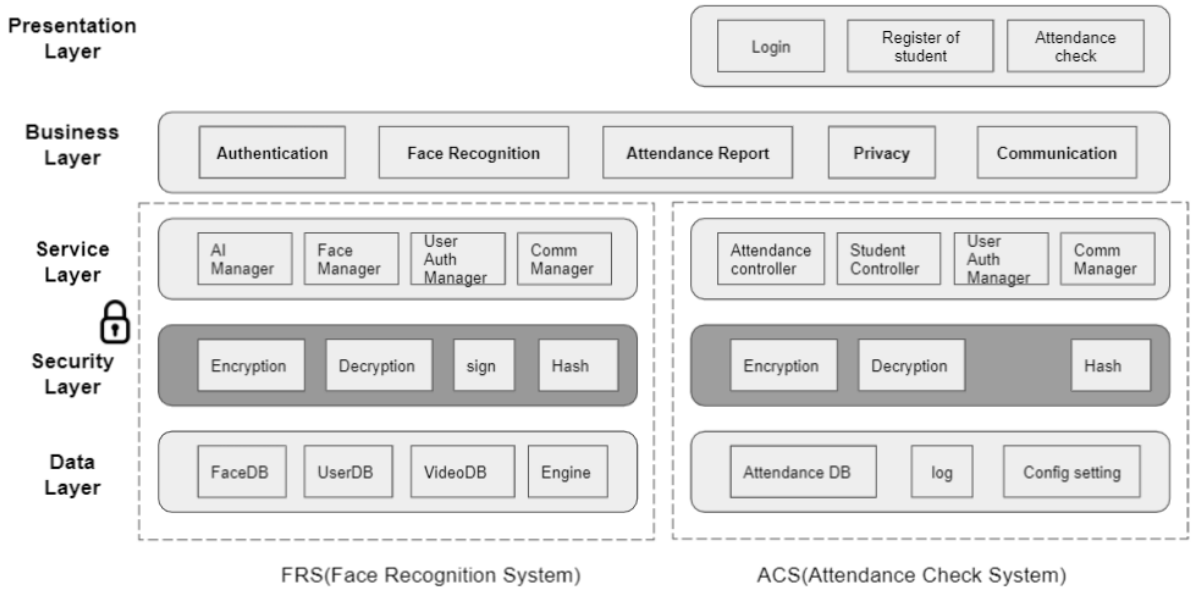


Figure 3 Static View

2.5.3.3. Architectural Alternatives

AACS believed that if data was leaked or compromised, it could cause big confusion and damage to CMU
How can we keep our data safe?
We thought that the most efficient way would be to manage the encryption key more securely.
We had to think about key management.

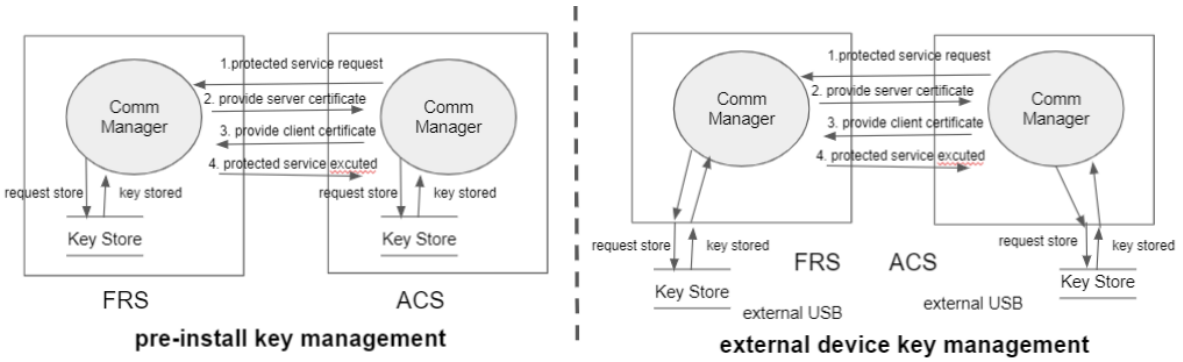


Figure 4 Architectural Alternatives

	Pre-installed key management	External Device key mangement
Pros	Easy to distribution Easy to key management	Update usb to update key Securely distribution Secure key storage
Cons	Update the whole program to update the key The key is easy to be exposed to risk	Physical usb key management Difficult to distribution

Architectural Decision :

Selected as an **external device key management** for key lifecycle management and secure key management

2.5.3.4. Physical View

The architecture reflecting the architectural decision is expressed below.
The additional physical device is expressed through the physical view as shown below.

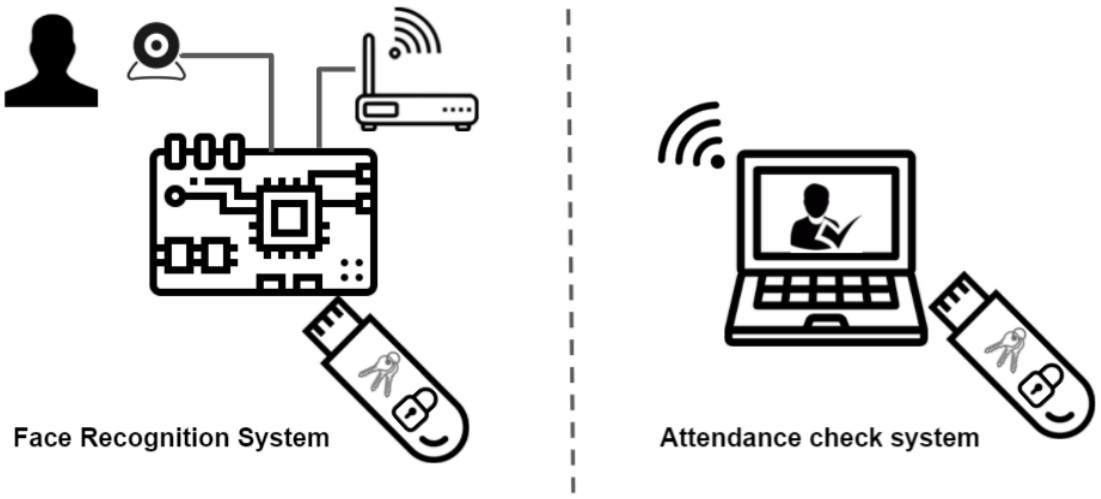
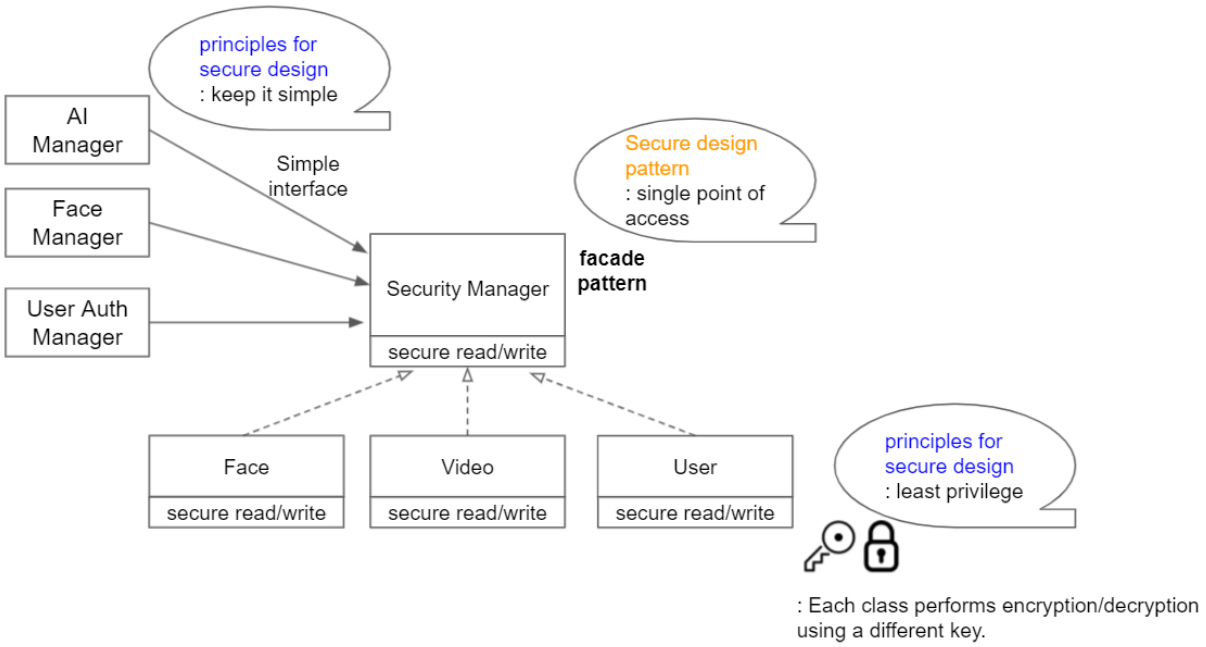
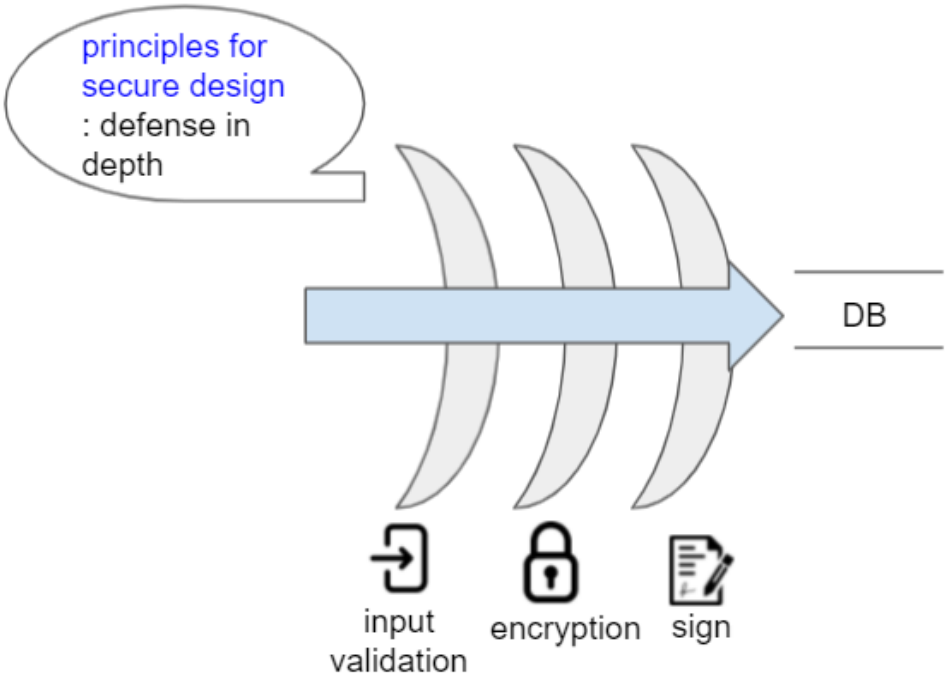


Figure 5 physical view

2.5.3.5. Secure Design Pattern





2.6. Implementation

2.6.1. Protocol Header

	TCP Header(54 bytes)	TCP payload (1460 bytes)
data_id	data_length	data
(2bytes)	(2bytes)	(1-1456 to N bytes) (truncation)

2.6.2. Protocol Details

FRS Module	Data ID(2byte) (Base + message id)		Data ID Name	Payload Data	Description
Face Mgr	0x1000	+0x0001	REQ_GET_FACES	none	request to get face data
		+0x0002	RESP_GET_FACES	[faces]	response with face data
		+0x0003	REQ_FACE_ADD	uid, num_of_faces	request to add face user data
		+0x0004	RESP_FACE_ADD	[faces]	response with face data
		+0x0005	REQ_FACE_DELETE	uid, fid	request to delete face user data
		+0x0006	RESP_FACE_DELETE	result	response with result
User Auth Mgr	0x2000	+0x0007	REQ_LOGIN	username, #password	request to user login
		+0x0008	RESP_LOGIN_OK	result	response login result
		+0x0009	RESP_LOGIN_FAILED	result	response login result
AI Mgr	0x3000	+0x000A	REQ_VIDEO_START	none	request to start live attendance
		+0x000B	REQ_VIDEO_END	none	request to stop live attendance
		+0x0010	RESP_VIDEO_FRAME	[frame]	response with frame data
		+0x0011	RESP_USER_ATTEND	[userlist]	response with attended user list
		+0x0012	REQ_STUDENT_LIST	none	request student list
		+0x0013	RESP_STUDENT_LIST	studentList	response with student list
		+0x0030	REQ_DISCONNECT	none	request to stop disconnet
		+0x0031	REQ_VIDEO_LIVE	videosource	request to change video source
		+0x0032	REQ_VIDEO_RECORD	videosource	request to change video source

2.7. Static Analysis

2.7.1. Static Analysis Summary

Static Analysis Tool	RATS-2.4			
Results (June 15, 2021)				
Report	*Severity*			
Folder	High	Low	Medium	총계
ControlAndDisplay	6	1		7
LgFaceRecDemoTCP_Jetson_Nano V2	4	9	6	19
총계	**10**	**10**	**6**	**26**
Issue of COUNTA	*Severity*			
Issue	High	Low	Medium	총계
EVP_DecryptUpdate			1	1
EVP_EncryptUpdate			1	1
fixed size global buffer	8			8
fixed size local buffer		6		6
memcpy		3		3
read			4	4
strlen		1		1
wsprintf	2			2
Total	**10**	**10**	**6**	**26**
Results (June 17, 2021)				
Severity of COUNTA	*Severity*			
Folder	High	Low	Medium	총계
ControlAndDisplay	6	2		8
LgFaceRecDemoTCP_Jetson_Nano V2	6	9	6	21
Total	**12**	**11**	**6**	**29**
Severity of COUNTA	*Severity*			
Issue	High	Low	Medium	총계
EVP_DecryptUpdate			1	1

EVP_EncryptUpdate			1	1
fixed size global buffer	10			10
fixed size local buffer		7		7
memcpy		4		4
read			4	4
wsprintf	2			2
Total	**12**	**11**	**6**	**29**

2.8. Testing

2.8.1. Test Case

TC ID	Category	Sub category	Brief Description	Precondition	Procedure	Expected Rusult	Result
AACS-TC-001	Login	Registered User Login with Real-time mode	Check the login of the user registered in AACS with Real-time mode.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected.	1. Enter the registered ID and password. 2. Select Real-time mode 3. Enter the login button.	"Welcome" Popup is shown.	Pass
AACS-TC-002		Registered User Login with Secure mode	Check the login of the user registered in AACS with Secure mode.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected.	1. Enter the registered ID and password. 2. Select secure mode 3. Enter the login button.	"Welcome" Popup is shown.	Pass
AACS-TC-003		Unregistered user login with Real-time mode	Check the login of the user unregistered in AACS with Real-time mode.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected.	1. Enter the unregistered ID and password. 2. Select Real-time mode. 2. Enter the login button.	"Login Fail." Popup is shown.	Pass
AACS-TC-004		Unregistered user login with Secure mode	Check the login of the user unregistered in AACS with Secure mode.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected.	1. Enter the unregistered ID and password. 2. Select secure mode. 2. Enter the login button.	"Login Fail." Popup is shown.	Pass
AACS-TC-005	Register of student	Add photo	Log in with the ID registered as student permission in AACS and register the face.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected. 4. Log in with registered student permission	1. Click the Add photo button. 2. Take photos	Photos are shown in UI	Pass
AACS-TC-006		Add 5 photo over	Log in with the ID registered as student permission in AACS and register the face.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected. 4. Log in with registered student permission	1. Click the Add photo button. 2. Take 5 photos 3. Try to add photos	No pop-up. Can not more add photos.	Pass

AACS-TC-007		Delete photo	Log in with the ID registered with student permission in AACS and delete the registered photo.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected. 4. Log in with registered student permission	1. Click the Delete photo button. 2. Try to delete one photo.	The registered photo is deleted.	Pass
AACS-TC-008		Delete 5 photo over	Log in with the ID registered with student permission in AACS and delete the registered photo.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected. 4. Log in with registered student permission	1. Click the Delete photo button. 2. Try to delete one photo and repeat it 4 times more.	The registered 5 photos are deleted.	Pass
AACS-TC-009		Finish/Logout	After logging in with the ID registered as student permission in AACS, click the finish button.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected. 4. Log in with registered student permission	1. Click the Finish button.	Go back to login scene.	Pass
AACS-TC-010	Attendance check	Live attendance check - Student's list	Log in to AACS with the id registered with admin permission and check if the student list is displayed.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected.	1. Login with the admin account	A list of registered students is displayed.	Pass
AACS-TC-011		Live attendance check - Start	When the start button is pressed in the AACS system, student attendance is checked in real time.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected. 4. Log in with registered admin permission 5. It is set to Live mode.	1. Login with the admin account 2. Press Start button	The attendance list is updated as the face of the person is detected from camera.	Pass
AACS-TC-012		Live attendance check - End	When the end button is pressed in the AACS system, student attendance is completed in real time.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected.	1. Live streaming from camera is shown.	1. Live streaming is ended. 2. Live streaming is recored as a video file.	Pass

				4. Log in with registered admin permission 5. It is set to Live mode.			
AACS-TC-013		Live attendance check - Clear	When the clear button is pressed in the AACS system, the student attendance result is cleared in real time.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected. 4. Log in with registered admin permission 5. It is set to Live mode.	1. Live streaming from camera is shown. 2. Press Clear button	The attendance list disappears.	Pass
AACS-TC-014		Past attendance check - Start	If it is set to past in the AACS system, student attendance proceeds through video when the start button is pressed.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected. 4. Log in with registered admin permission 5. It is set to Past mode.	1. The video can be selected and loaded. 2. Press Start button.	1. The video image is shown. 2. The attendance list is updated.	Pass
AACS-TC-015		Past attendance check - End	If it is set to past in the AACS system, student attendance is completed through video when the end button is pressed.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected. 4. Log in with registered admin permission 5. It is set to Past mode.	1. The selected video is played. 2. Press End button.	1. Video play ends. 2. The update of attendance list is done.	Pass
AACS-TC-016		Past attendance check - Clear	If it is set to past in the AACS system, the student attendance result is cleared through video when the clear button is pressed.	1. FRS is operating normally. 2. ACS/FRS connected to the network. 3. The USB key of ACS/FRS is connected. 4. Log in with registered admin permission 5. It is set to Past mode.	1. The selected video is played. 2. Press Clear button	The attendance list disappears.	Pass
AACS-TC-017	Security for Client	USB Key for Client	In the AACS system, only authorized PCs should be able to access the	1. No USB Key for client is connected. 2. Go to Login scene in Client program.	1. Press Login button.	Warning message box opened. "Unable Acess USB Key. Not	Pass

			server. The USB Key for client PC is required.			Connected	
AACS-TC-018		USB Key for Client	In the AACS system, only authorized PCs should be able to access the server. The USB key for client PC is required.	1. USB Key for client is connected 2. Go to Login scene in Client program.	1. Press Login button.	"Welcome" Popup is shown.	Pass
AACS-TC-019		Connection configure check	Connection configuration information should not be tampered except the administrator	1. Modifiy clientconf.bin file 2. USB Key for client is connected 2. Go to Login scene in Client program.	1. Press Login button.	Warning message box opened. "Configuration file is missing or changed. Not Connected"	Pass
AACS-TC-020		Connection configure check	Connection configuration information should not be tampered except the administrator	1. Use the original clientconf.bin file 2. USB Key for client is connected 2. Go to Login scene in Client program.	1. Press Login button.	"Welcome" Popup is shown.	Pass
AACS-TC-021	Security for Server	USB Key for Server	In the AACS system, the USB key for Server which is regarded as secure storage is required	1. No USB Key for Server is connected. 2. Server program is not yet executed.	1. Execute Server program.	Program exit. "Program exit. check usb keys"	Pass
AACS-TC-022		USB Key for Server	In the AACS system, the USB key for Server which is regarded as secure storage is required	1. USB Key for Server is connected. 2. mount /mnt/usb/cert/rootca.crt 3. Server program is not yet executed.	1. Execute Server program.	No error message is shown.	Pass
AACS-TC-023		Engine file hash check	The tempering attempt for a model file which is used in facenet and mtCNN should be detected and noticed.	1. Make some changes in engine data, uff, caffemodel, prototxt 2. Server program is not yet executed.	1. Execute Server program.	Program exit. "Program exit. check facenet and mtcnn model files."	Pass
AACS-TC-024		Engine file hash check	The tempering attempt for a model file which is used in facenet and mtCNN should be detected and noticed.	1. No change in the original engine, uff, caffemodel, prototxt files 2. Server program is not yet executed.	1. Execute Server program.	No error message is shown.	Pass

AACS-TC-025		DB should be encrypted	DB should be encrypted.	1. Get UserDB, FaceDB, VideoDB Files from Server	1. Check strings from DB Files
					$ strings dbfile
					2. Check jpeg signature(52 49 46 46)
					$ grep -obUaP "\x52\x49\x46\x46" dbfile
					3. Check avi signautre(FF D8 FF E0)
					$ grep -obUaP "\xff\xd8\xff\xe0" dbfile
						No valid string is shown.
						No signaure is shown or meaning less bytestream appeared
							Pass(UserDB)

AACS-TC-026		DB should be signed and verified	The signed file should be created every time DB is saved.	1. Get DB.bin, DB.sign file from server 2. Get rootca.crt from USB Key	1. Verify with OpenSSL 2. set date +6 month 3. Verify with OpenSSL 4. set date -6 month 5. Verify with OpenSSL	"Verification successful" at step 1 "Verification failure" at step 3 "Verification successful" at step 5	Pass(UserDB)
AACS-TC-027	Security for Communication	Support TLS and promised ciphersuite	TLS1.3 is supported. TLS_AES_128_GCM_SHA256 is supported as ciphersuite.	1. Go to Login scene in Client program. 2. Wireshark application is monitoring the network packets.	1. Press Login button. 2. Check TLS handshake packets	Login success Check wireshark description: - TLSv1.3 Record Layer: Handshake Protocol: Client Hello - Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)	Pass
AACS-TC-028		Server-Client mutual authentication is required	The server and client must exchange their certificates.	1. Go to Login scene in Client program. 2. Wireshark application is monitoring the network packets.	1. Press Login button. 2. Check TLS handshake packets	"Welcome" Popup is shown. Check TLS handshake packet length. - Server Hello Packet: more than 1269 Byte - Next Packet(C->S): more than 1269 Byte	Pass

For the remaining test cases, refer to Team3_AACS_Testcase.

2.8.2. Test Case Summary

Test cases are organized in Team3_AACS_Testcase for testing the implemented functions. The result of checking the current implementation state through the test case is as follows.

The test didn't pass 100%, but we think it's the result of trying our best during the three weeks of development.

Item	Description
Test Scope	AACS Requirements
Requirement Test Coverage	**80%**
Test Level	**System Test**
Test Method	Manual Test
Number of Test Cases	49
Pass	45
Fail	4
Pass Rate	**91.8 %**

2.9. Vulnerability Analysis

2.9.1. Vulnerability Analysis Summary

COUNTA of Severity	*Severity*			
OSS Name / Version	High	Low	Medium	Grand Total
NVIDIA Tegra kernel v4.9		1	2	3
OpenCV v4.1.1			1	1
OpenSSL v1.1.1	2	2	8	12
Grand Total	**3**	**2**	**11**	**16**

The Team3_Open_Source_Vulnerability_Report document was created by reviewing vulnerability for the open source used while developing AACS.

2.10. Cryptographic Algorithms

2.10.1. Secure communication with TLS 1.3

Category	Recommendations	Applied to our project	Reason for selection
SSL/TLS version	TLS 1.2 or TLS 1.3	TLS 1.3	The latest version of TLS Don't need to consider backward compatibility The highest level of security
Cryptographic library for TLS	GnuTLS OpenSSL wolfSSL	OpenSSL v1.1.1 - v1.1.1k (client) - v1.1.1 (server)	Widely used in industries for the commercial products
Cipher suites	Recommended for TLS 1.3 : TLS_AES_128_GCM_SHA256 TLS_AES_256_GCM_SHA384 TLS_AES_128_CCM_SHA256	TLS_AES_128_GCM_SHA256	AES-128 for performance
Signature keys for certificates	RSA : 2000 bits ~ (~2023) 3000 bits ~ (2024~2027+)	RSA-2048 bits	RSA-2048 bits for performance

2.10.2. Security Manager

1. Handle all cryptographic keys and certificates
2. Provide all security functions for the processes based on openssl (v1.1.1)
- hash (sha 256)
- encryption/decryption (AES-128)
- signing/signing verification (RSA-2048)
- Set/free network context for TLS
3. Applied Facade pattern (Hide complexity / Provide simple interface)

2.10.3. Cryptographic Algorithms Applied

Category	Recommendations	Applied to our project	Reason for selection
Block ciphers	AES-128, AES-192, AES-256	AES-128	Choose the shortest key length for performance

Mode of operation for block ciphers	CCM (Counter with Cipher Block Chaining Message Authentication) GCM (Galois/Counter Mode) CBC (Cipher Block Chaining) CTR (Counter Mode)	CBC	The most secure mode among the block cipher modes
Hash functions	SHA-256, SHA-512/256, SHA-384 and SHA-512 SHA3-256, SHA3-384, SHA3-512	SHA-256	Most widely used. Fast and strong enough for most purposes.
Digital signatures	RSA, DSA, ECDSA, ECKDSA, ECGDSA, XMSS+ or LMS	- RSA-2048 bits (Self-signed certificate)	Most widely used.

2.10.4. Key Management

Category	Recommendations	Applied to our project	Reason for selection
Key distribution	The generated keys shall be transported (when necessary) using secure channels and shall be used by their associated cryptographic algorithm within at least FIPS 140-2 compliant cryptographic modules.	All cryptographic keys and client/server CAs are distributed via USB (assuming that USB is a kind of a secure storage). Root CA is distributed with the developed software.	Decide to use USB instead of FIPS compliant module due to the time limitation.
Key storage	Ensure all keys are stored in cryptographic vault, such as a hardware security module (HSM) or isolated cryptographic service	All cryptographic keys are stored in USB.	Assume that USB is a kind of secure storage like HSM.
Key revocation	The details for key revocation should reflect the lifecycle for each particular key	Do not consider this scenario.	Key lifecycle management and revocation scenario can not be implemented due to the time limitation.

2.10.5. Crytoperiod

No.	Key Type	Cryptographic keys stored in server-side USB	Cryptographic keys stored in client-side USB	Cryptoperiod Recommended	
				Originator-Usage Period (OUP)	Recipient-Usage Period
1	Private signature key	Private signature key for signing Video DB Private signature key for signing User DB Private signature key for signing Face DB		1 to 3 years	-
2	Public signature-verification key	Public signature-verification key for Video DB Public signature-verification key for User DB Public signature-verification key for Face DB Root CA certificate	Client certificate Root CA certificate	Several years (depends on key size)	
4	Private authentication key	Server key	Client key	1 to 2 years	

5	Public authentication key	Server certificate Root CA certificate	Client certificate Root CA certificate	1 to 2 years	
6	Symmetric data encryption key	Symmetric data encryption key for encrypting Video DB Symmetric data encryption key for encrypting User DB Symmetric data encryption key for encrypting Face DB		Up to 2 years	Up to OUP + 3 years

Applied to our project

- Root CA certificate cryptoperiod : 365 days
- Client/Server certificate cryptoperiod : 90 days

2.10.6. References

1. BSI TR-02102-1 : Cryptographic Mechanisms: Recommendations and Key Lengths,
BSI Technical Guideline, Mar 24, 2021.

2. BSI TR-02102-2 : Cryptographic Mechanisms: Part 2 - Use of Transport Layer Security (TLS),
BSI Technical Guideline, Mar 12, 2021.

3. NIST SP 800-131A Rev. 2: Transitioning the Use of Cryptographic Algorithms and Key Lengths,
NIST Special Publication, Mar 21, 2019.

4. NIST SP 800-57 Revision 5: Recommendation for Key Management : Part 1 - General,
NIST Special Publication, May 4, 2020.

5. NIST SP 800-133 Rev.2 : Recommendation for Cryptographic Key Generation,
NIST Special Publication, June 2020.

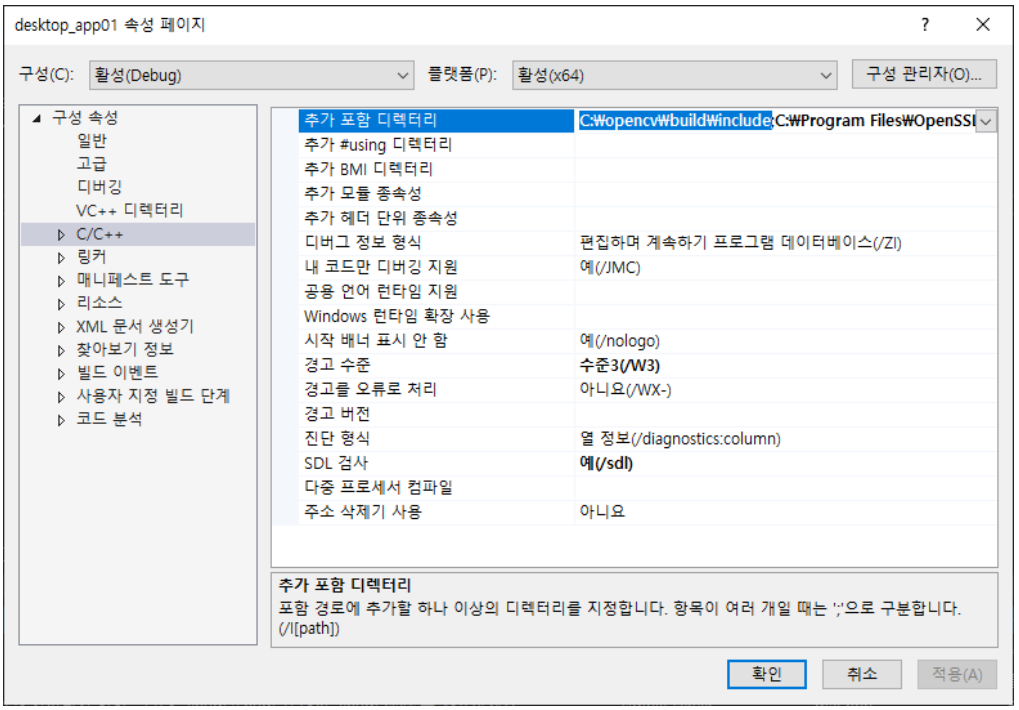
2.11. Setup Manual

2.11.1. PC Client Setup

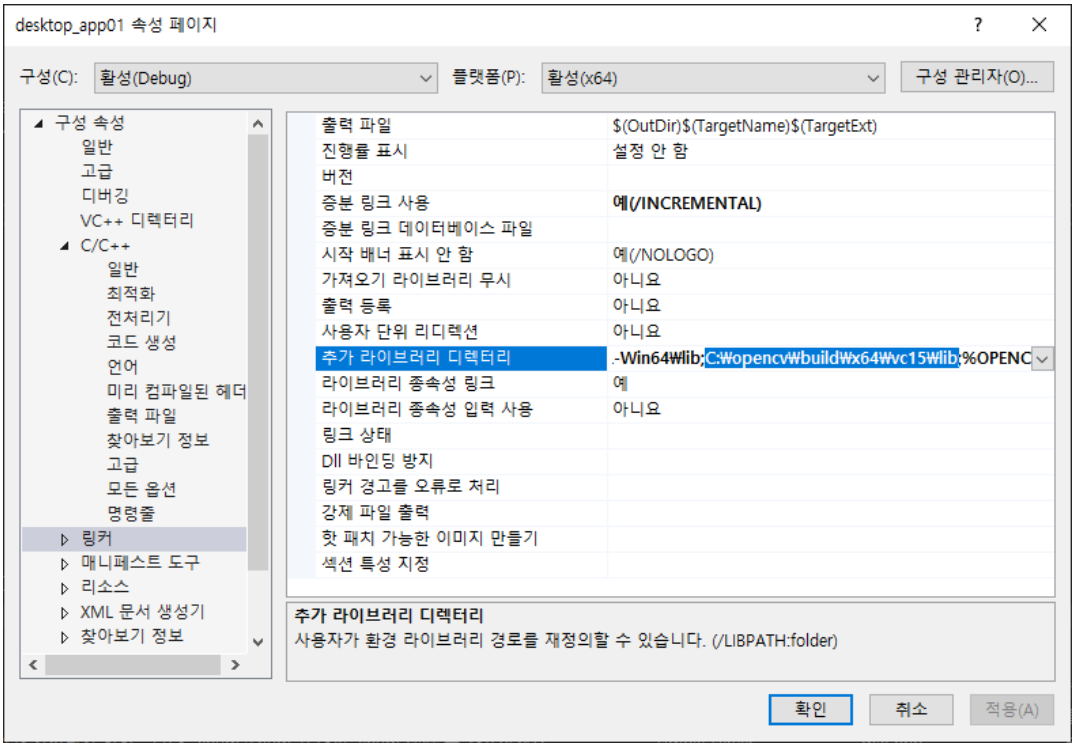
Download Source Code
https://github.com/SWSS2021Team3/ControlAndDisplay

- desktop_app01 : PC Client Source Code Directory
- gtest_sample01 & simple-server : Source code directory for internal testing

OpenCV Setup for MS Visual Studio
Add the installed opencv\build\include path to the project's property page -> C/C++ -> additional include directories



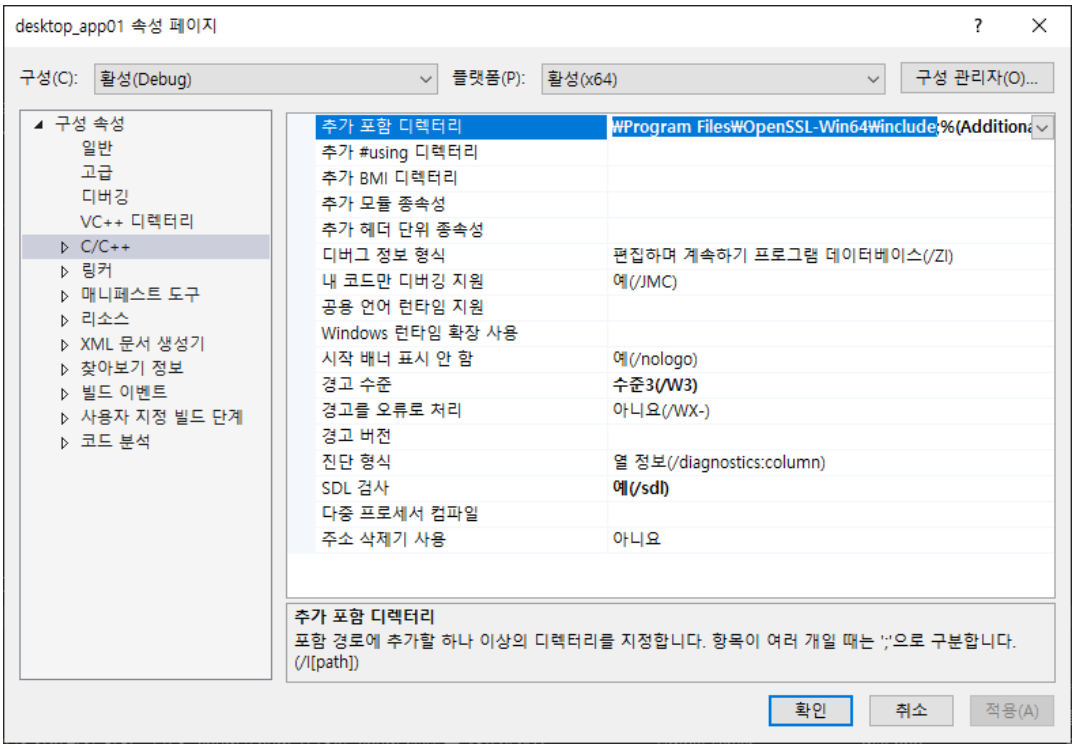
OpenCV Setup for MS Visual Studio
Add the installed opencv\build\x64\vc15\lib path to your project's property page -> Linker -> Additional include directories



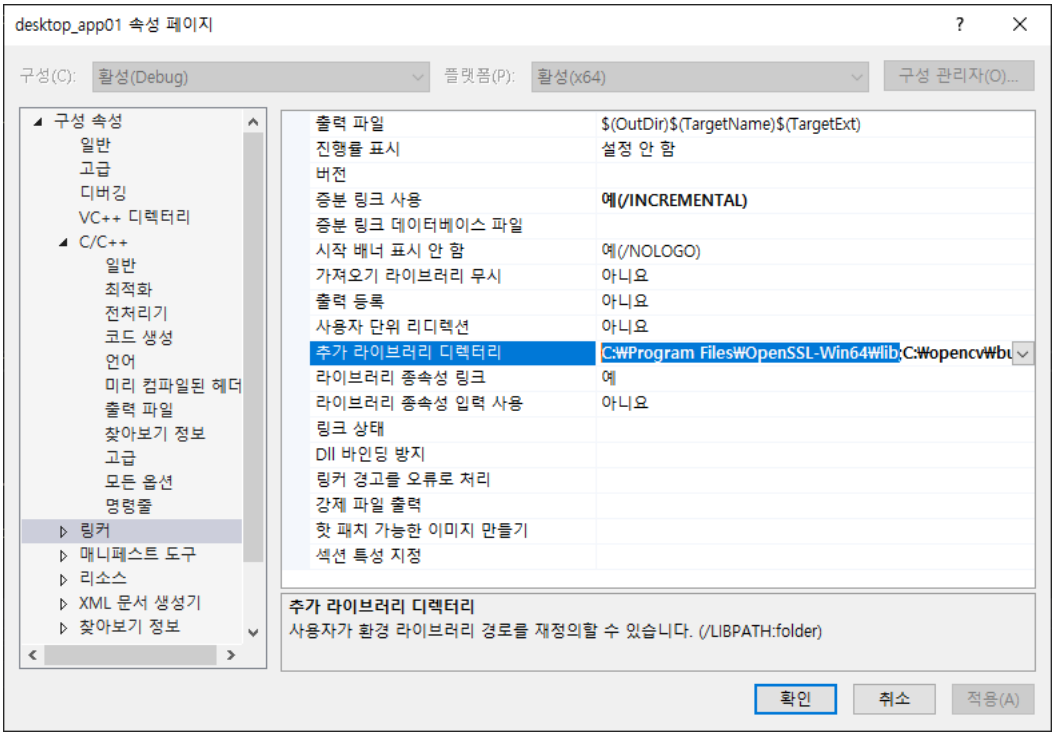
OpenSSL Download / Install
https://slproweb.com/products/Win32OpenSSL.html

Download and install Win64 OpenSSL v1.1.1k (63MB Installer)
Default installation path: C:\Program Files\OpenSSL-Win64

OpenSSL Setup for MS Visual Studio
Add the installed OpenSSL-Win64\include path to your project's property page -> C/C++ -> Additional Include Directories



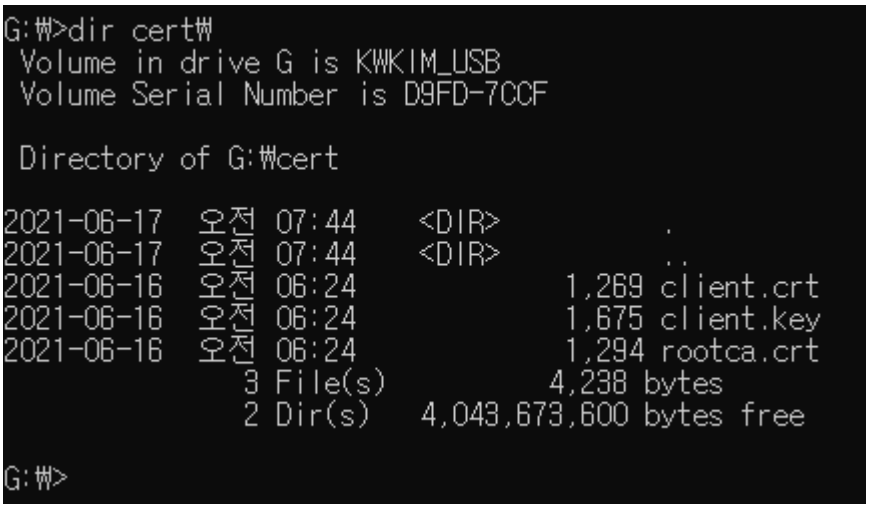
OpenSSL Setup for MS Visual Studio
Add the installed OpenSSL-Win64\lib path to your project's property page -> Linker -> Additional Include Directories



USB Key Setup(If you need USB Drive, then ask to TEAM 3)

Download USB Key
https://github.com/SWSS2021Team3/USB_Key

Copy ClientKey*.* to USB Flash Drive
rootca.crt file should be exist on “USB_FLASH_DRIVE:\cert\rootca.crt”



Account Information to log in

ID	Password	Authorization
admin	qorvjdlswmd	Admin
kyuwoon.kim	rlarbdns	User
gyeonghun.ro	shrudgns	User
wonyoung.chang	wkddnjsdud	User
soohyun.yi	dltngus	User
hyejin.oh	dhgPwls	User
hyungjin.choi	chlgudwls	User
vibhanshu.dhote	qlqmgkstb	User
cliff.huff	zmfflvm	User

1.1.1. Server Setup

Download Source Code
https://github.com/SWSS2021Team3/LgFaceRecDemoTCP_Jetson_NanoV2

Project Build

```
/LgFaceRecDemoTCP_Jetson_NanoV2$ mkdir build
/LgFaceRecDemoTCP_Jetson_NanoV2$ cd build
/LgFaceRecDemoTCP_Jetson_NanoV2/build$ cmake ..
/LgFaceRecDemoTCP_Jetson_NanoV2/build$ make
```

Project Run

```
/LgFaceRecDemoTCP_Jetson_NanoV2/build$
./LgFaceRecDemoTCP_Jetson_NanoV2 [PORT]
```

OpenSSL Library Setup

Library install

```
$ sudo apt-get install libssl-dev
```

USB Key Setup (If you need USB Flash Drive, then ask to TEAM 3)

Mount to Jetson Nano

```
$ sudo mkdir /mnt/usb
$ sudo mount -t vfat /dev/sda1 /mnt/usb
```

Download USB Key
https://github.com/SWSS2021Team3/USB_Key

Copy ServerKey*.* to /mnt/usb/
rootca.crt file should be exist on “/mnt/usb/cert/rootca.crt”

```
lg@LgFaceRecProject:~$ ls -l --file-type /mnt/usb/
total 32
drwxr-xr-x 2 root root 16384 Jun 16 04:12 cert/
drwxr-xr-x 2 root root 16384 Jun 16 04:12 db/
lg@LgFaceRecProject:~$ ls -l --file-type /mnt/usb/cert/rootca.crt
-rwxr-xr-x 1 root root 1294 Jun 16 04:12 /mnt/usb/cert/rootca.crt
lg@LgFaceRecProject:~$
```

Change engine and model files. (If you need. All the engine files are check HASH.)

Get hash

```
$ openssl sha256 facenetModels/*
SHA256(facenetModels/facenet.engine)= 71493446240e3f9286437c5a0baab41aae6a1e47ddbcb21a24079440ba0e5d86
SHA256(facenetModels/facenet.uff)= 7049d18ad472b535ccd022edf80707d6598ad81472028f1c0024274524ff6ce4
SHA256(facenetModels/README.md)= c7337e9a56c2228229728eb09b823f85025e21387e228e32d3fb8e0d8658d7af

$ openssl sha256 mtCNNModels/*
SHA256(mtCNNModels/det1_relu1.engine)= 3c26255262c050185c5fa45521a96fedda59f635457b5c8795268a749f7a4c70
SHA256(mtCNNModels/det1_relu2.engine)= 9aab5f7788dc385391324c1bf6e51e92c5a2fdbf2f337490bf6cbbb063032a00
SHA256(mtCNNModels/det1_relu3.engine)= 0c7a81f29c930acb1ce67ca64a107b74ff3917760a8e7bdcd45343d6db4f022c
SHA256(mtCNNModels/det1_relu4.engine)= 63866e8b088bd3e12fb4122dc569ee78e6b8c769dd529ffe62fb7f938d6659d5
SHA256(mtCNNModels/det1_relu5.engine)= 749a2f60a17677f3052cfca78fc6040d1b0a540766acfa70777203ed2f59f463
SHA256(mtCNNModels/det1_relu6.engine)= e4ac1e0b5ff0383fb3fd612826d0ac94fa67f094350c6439ab07b88d12fd6df9
SHA256(mtCNNModels/det1_relu7.engine)= fafbf0d0bd89ee02929940878b45edb80a0497bb99f881aab7b618a511434877
SHA256(mtCNNModels/det1_relu.caffemodel)= b46dbcf61b858c1ec67ffdf86b805a050e25b095b5f172b8bdedd48149f2dbbf
SHA256(mtCNNModels/det1_relu.prototxt)= 6e1bedd5b73017623249445b43a1ca07eeb68343ff310bce4348bda9e3a50567
SHA256(mtCNNModels/det2_relu.caffemodel)= 053a4445c392878649aeed457ea1f3f7f5a1e23bfe29cb038c451131ed96a469
SHA256(mtCNNModels/det2_relu.engine)= 2253f2a34e568d0a9c033ae5028e9339918fff4d4c2832bf351d28f06b5b3ac5
SHA256(mtCNNModels/det2_relu.prototxt)= 3d6986b38f98954f57be8108f1b09794e2890906f2006526172139c3b5a2bff6
SHA256(mtCNNModels/det3_relu.caffemodel)= f5bf43cd05feea8fb5f7250dcc610065308e66f44b1fb2cd956bfcd43ae58c79
SHA256(mtCNNModels/det3_relu.engine)= be3c6934f2c112f34f20a3a99f1ee8bb7d272894a2fa772c99fcacf1ae419507
SHA256(mtCNNModels/det3_relu.prototxt)= 59f75d1ca76a78333646ff7d6c92e5866f187831f3579345ab7b62406efccf7e
SHA256(mtCNNModels/README.md)= eee30b9ebf9c7946ec60811d7e969aa8cf09921aecd13a7b9a85043ce2988259

Appy hash value(right hand of '=' ) to SecurityManager::readKey() at src/SecurityManger.cpp File.
```

1.1.2. Admin Setup

Admin could make connection information(IP/Port/SecurePort)

```
* prepare for sign
openssl.exe
clientconf.bin  (Ip, port, secure port are written each line.)
clientconf.crt  (https://github.com/SWSS2021Team3/USB_Key)
clientconf.key (https://github.com/SWSS2021Team3/USB_Key)

> type clientconf.bin
192.168.0.106
5000
5010

> openssl cms -sign -in clientconf.bin -signer clientconf.crt -inkey clientconf.key -outform DER -out clientconf.sign -binary

> copy clientconf.bin clientconf.sign [client application directory]
```

1.1.3. Deployment Setup

Download deployment
https://github.com/SWSS2021Team3/ProjectDocuments
It exists in the deploy subdirectory of ProjectDocuments

ACS : Attendance Check System (PC)
unzip ACS.zip
run ACS.exe

FRS : Face Recognition System (Jetson Nano)
Install
cat FRS.tar* | tar xvf -
unzip FRS.zip
cd deploy

Run Camera mode (has some problem with closing camera)
./run.sh

Run Play Record mode (can be switched to Camera mode)
./run-mov.sh

2.12. Remaining works

2.12.1. Development Part

1. video file / photo(jpg) file encryption / decryption is not complete.
2. Input validation (username, password, record filename)
3. Abnormal disconnect(timeout)
4. Exception handling : GStreamer camera open/close fail
5. Password change
6. 2 Factor Authentication
7. Certificate revocation
8. Certificate update
9. Certificate ocsp
10. Intermediate CA
11. OTP authentication(for login or issue new key)
12. Captcha
13. Oauth
14. More strict file check for DB files
15. change DB encrypt password
16. TLS renegotiation

18. DTLS for video streaming
19. Secure aes key expansion
20. Hardware id or serial number check
21. Key or Certificate generation with API
22. Prevent keyboard hooking at PC application
23. IP filtering
24. Firewall
25. Logging system

2.12.2. Attendance Check System(PC Application)

1. User ID, Password input validation
2. Complex password check
3. System log
4. Captcha
5. 2-FA Authentication
6. OAuth
7. Prevent Keyboard Hooking
8. Password change

2.12.3. Face Recognition System(Jatson nano part)

1. Video file / photo(jpg) file encryption / decryption is not fully implemented
2. Abnormal disconnect(timeout)
3. Exception handling : GStreamer camera open/close fail

4. More strict file check for DB files
5. change DB encrypt password
6. Multi user concurrent login support
7. Real secure storage for private key
8. Secure aes key expansion

2.12.4. Common System

1. Real secure storage for private key
2. TLS renegotiation
3. DTLS for video streaming
4. Hardware id or serial number check

2.12.5. Certificate

1. Certificate revocation
2. Certificate update by online
3. Certificate ocsp
4. Intermediate CA
5. OTP authentication for Private Key redistribution
6. Key or Certificate generation with API

2.12.6. Jatson Nano Ubuntu

1. IP filtering
2. Firewall
3. Newer version of OpenSSL

2. Strength / Weakness

2.1. Strength

In designing security, all members proceeded together according to all possible processes.

Since all members participated in the requirements and architecture design and proceeded, I think the output for that part is a strength compared to other teams.

In addition, all processes from requirements to testing were carried out according to the process presented in the software development process, and the output was made.

A lot of effort was put into testing as well. We tried to elaborate on the test case writing, and we tried to build an efficient test environment through the simulator when testing.

2.2. Weakness

The part that I thought was lacking in the development process was implementation.
In a situation where the development period was limited to 3 weeks, there was a time limit because we had to implement the remaining time because we focused on the requirements and design.

It seems that various security factors were not reviewed from the perspective of applying security factors to the system.

We did not give feedback on the architecture that was designed through the requirements and the little bit improved architecture through implementation. If we had a little more time, we think we would have made an effort to create a more structured architecture through feedback on the initially designed architecture.

Finally, we have regrets about the test. we wanted to try various tests according to the unit, integrate, and system levels for each test level. However, according to the schedule, only system level tests were carried out.

3. Lesson & Learned

3.1. What all members felt

1. It was a good opportunity to learn about the process of enforcing security in architecture.
2. The security was reflected in the design from the beginning, so the implementation had been well.
3. Lack of time was the biggest constraint, but we were able to overcome it through teamwork.
4. When faced with a problem in the project, I was able to learn how to apply to this project and to make a decision with discussion to solve the problem.
5. It's good to know the usage of openssl and learn from team mates about security manager implementation.
6. From design to deployment, I felt that secured software had a lot to consider and took time.
7. There were too many requirements to implement, but proper decision on what can be done in within given time help a lot to reach to a good shape.