DIAMOND Portfolio

COMP60721 Systems Governance

Group 1: Yalow

Group Component: Secure Architecture

Umar Farooq	10326509
Lujin Li	10815865
Jiaxin Guo	10856460
Shubham Agarwal	10918790
Zirui Wang	10868665
Jinliang Sun	10907267
Xinran Zhao	10855951
Shuhao QI	10822869

Table of Contents

1 Introduction

For this group assignment a case study of a DIAMOND (Driver Identification After Motoring Offence using Numerous Data sources) system that can identify drivers that are driving a speeding vehicle. From working together, we were able to create Architecture Schematics of the DIAMOND system also exploring the Performance/Conformance of the system by creating a dashboard and creating a plan to ensure that the system is trustworthy.

For this group component our team divided the tasks required into small sections for each of us to do, individually and collaboratively. Below a table can be seen on what each member of our group contributed to the assignment. (As shown in Table1.1)

Name	Assigned tasks and responsibilities
Umar Farooq	Team Leader, System Requirements/Risk, Use/Misuse Diagram, Scorecard
Jinliang Sun	System Requirements, Use/Misuse Diagram, Conformance: Availability
Lujin Li	System Flow Chart and Network Structure, Risk, Dashboard, Format, Revise
Shubham Agarwal	Responsibility Identification, Class Diagram, Scorecard, People-device symbiosis
Shuhao Qi	Data Flow Planning, Risks Analysis, Conformance: Security
Jiaxin Guo	System Requirement, Class Diagram, Conformance: Resilience
Xinran Zhao	Responsibility Identification, Risks Analysis, Conformance: Safety
Zirui Wang	Data Flow Planning, Risks Analysis, Conformance: Reliability

Table1.1 - Team member responsibilities

2 Responsibilities

From discussion with our team, we devised the Owners, Customers and Actors of our system which can be seen in the table below.

In the specification it was mentioned that the Home Office commissioned the creation of the DIAMOND system, so they are therefore presumed to be the owners of the system. (As shown in Table 1.2)

Owners	Home Office (Government)
Customers	Local Authority Police DVLA
Actors	Driver Police ANPR (inc. Petrol Station/Car Park cameras) Speed Camera DVLA Database Passport Service Database Home Office Database

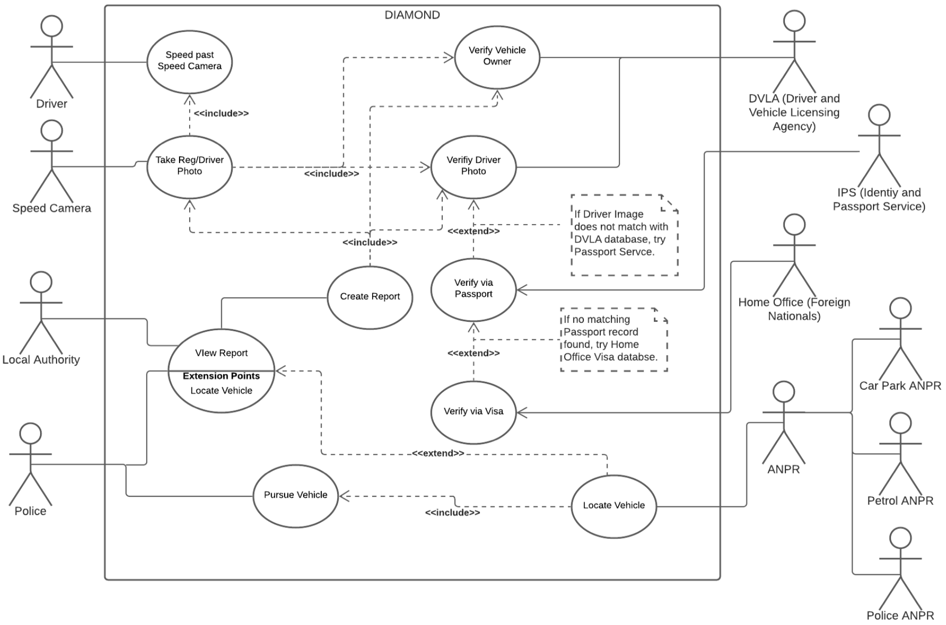
Table1.2 - DIAMOND system responsibilities

3 Architecture Schematics

This section shows core designs of the system in five different diagrams.

3.1 Use-Case Diagram

The use case diagram (As shown in Graph3.1) demonstrates how the users may interact with our system.



Graph3.1 - Use-case diagram

Firstly, the speed camera, at the left-top corner of the diagram, will detect the cars that are out of speed and will take photos of the driver and car plate.

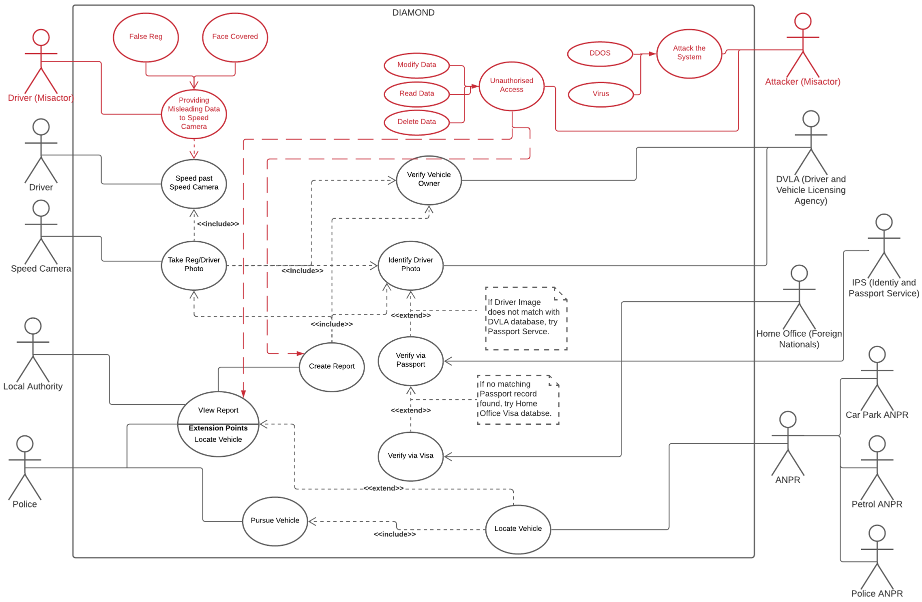
Then our system will use the information to create an offence report, cooperating with DVLA, Identity and passport service and Home Office, to identify the driver step by step, through verifying the photo, passport and visa.

Next the report is sent to Local authority and police office so that they can view the reports through our system.

What’ s more, they can locate a vehicle involved in an offence through ANPR records of difference places so that they can pursue the driver.

3.2 Misuse-Case Diagram

The Misuse-Case diagram below (As shown in Graph3.2) has been created to demonstrate what is expected to go wrong in the DIAMOND system.

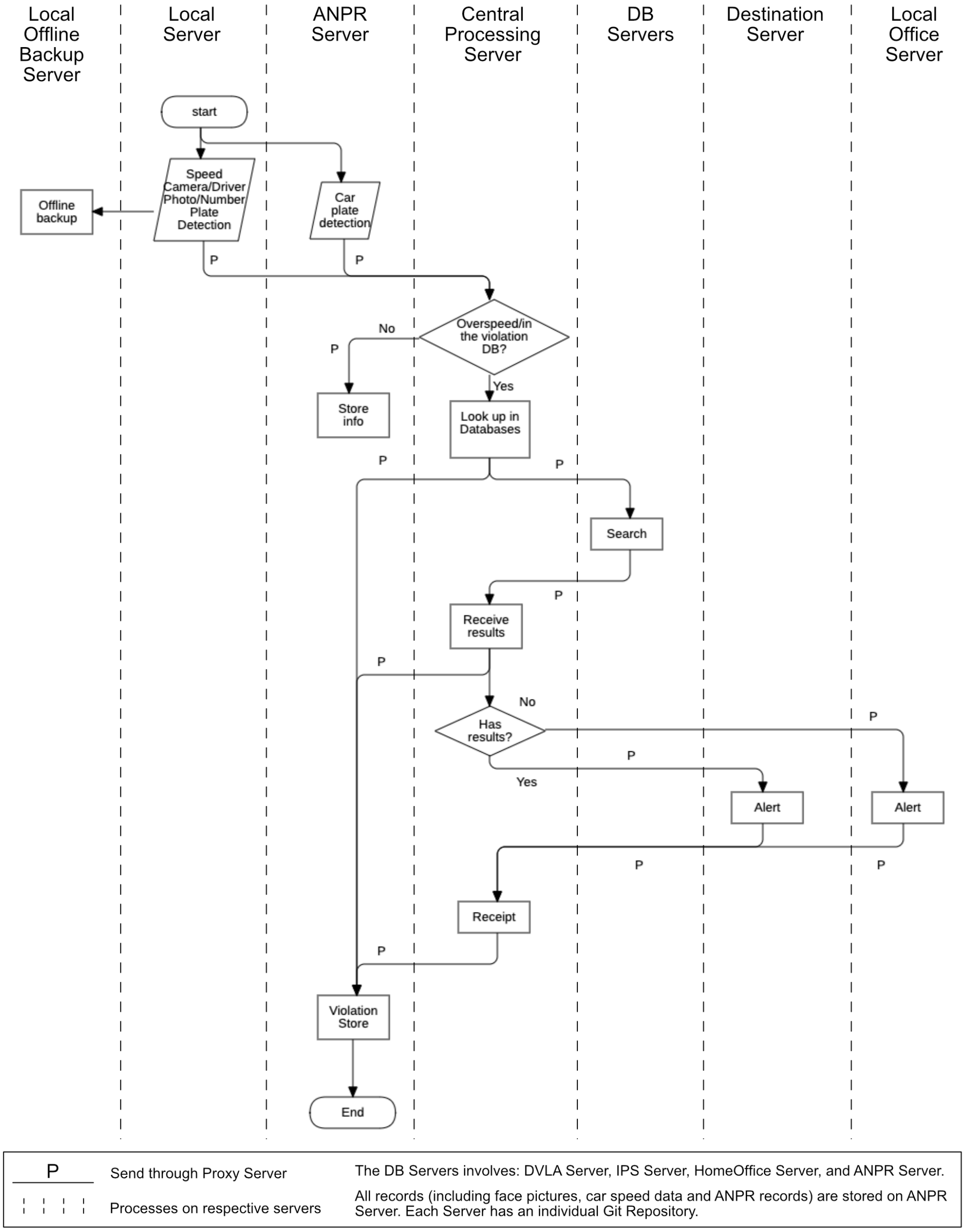


Graph3.2 - Misuse-case diagram

The diagram demonstrates malicious miss actors in the system and how they could cause intentional damage with unauthorized access and attacks to the system. The diagram above also demonstrates how a normal actor like the Driver also has the potential to be a miss actor where they could intentionally or unintentionally cause the speed camera to capture incorrect/misleading data.

3.3 Flow Chart

The flow chart in Graph3.3 shows the general dataflow of the DIAMOND system.


Local Offline Backup Server
Local Server
ANPR Server
Central Processing Server
DB Servers
Destination Server
Local Office Server
start
Speed Camera/Driver Photo/Number Plate Detection
Offline backup
Car plate detection
P
P
Overspeed/in the violation DB?
No
P
Store info
Yes
Look up in Databases
P
P
Search
P
Receive results
P
No
Has results?
P
P
Yes
Alert
Alert
P
P
Receipt
P
Violation Store
End
P
Send through Proxy Server
Processes on respective servers
The DB Servers involves: DVLA Server, IPS Server, HomeOffice Server, and ANPR Server.
All records (including face pictures, car speed data and ANPR records) are stored on ANPR Server. Each Server has an individual Git Repository.

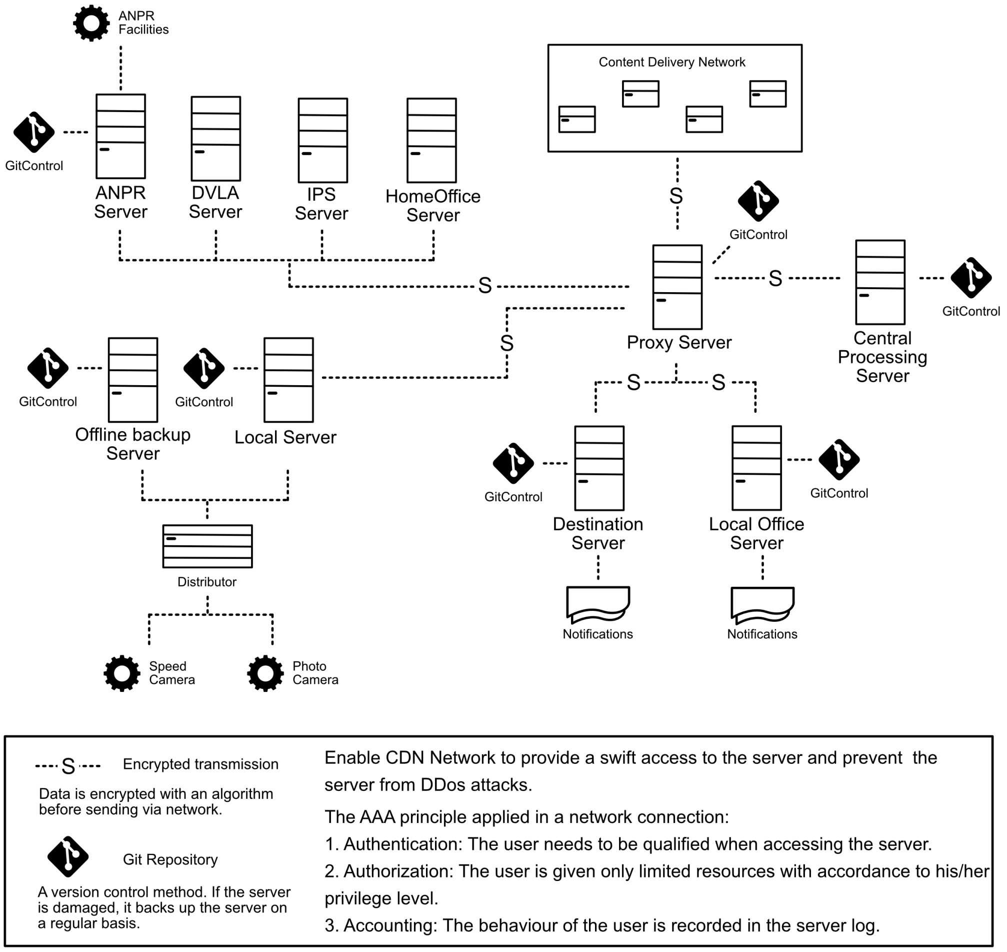
Graph3.3 - Flow chart

The system is designed to work 24/7, starting by two data entries: the photo & speed cameras and the ANPR network. These facilities automatically capture passing-by vehicles’ information and send it to the central processing server.

The central processing server is responsible to tell whether a vehicle has violated speed law, and identify the driver inside if it was out of speed. The server will use the photo captured by a photo camera to help look the driver up in several databases. If there is a record of the driver in the database, then a penalty notification will be sent to the police office and the driver himself. If we cannot find any record within our databases, the notification will go to the local office to warn them there might be an illegal driver on the road.

3.4 Network Architecture

The Network architecture diagram shows an overall design of the architecture of the system (As shown in Graph3.4).



Data is encrypted with an algorithm before sending via network.

A version control method. If the server is damaged, it backs up the server on a regular basis.

Enable CDN Network to provide a swift access to the server and prevent the server from DDos attacks.

The AAA principle applied in a network connection:
1. Authentication: The user needs to be qualified when accessing the server.
2. Authorization: The user is given only limited resources with accordance to his/her privilege level.
3. Accounting: The behaviour of the user is recorded in the server log.

Graph3.4 - Network architecture

So instead of possessing only one or two servers, the DIAMOND system has deployed 5 servers in the entire system.

The difference is that we set an individual server for the ANPR facilities and we store the most frequently used data on that server as well. We need this server to store all the violation records, we want the ANPR facilities to work 24/7, so once the ANPR detects a car with an unsolved speed crime, it can use the inbuilt server to compare the records, and immediately send the alert to the office, attached with its location.

And everything decision-related is processed by a central processing server, to ensure a high efficiency of data processing and decision management. The Central processing server is responsible for decisions like: telling if a driver is driving out of speed, or whether the driver's information is in the database and where the result would be stored or sent to.

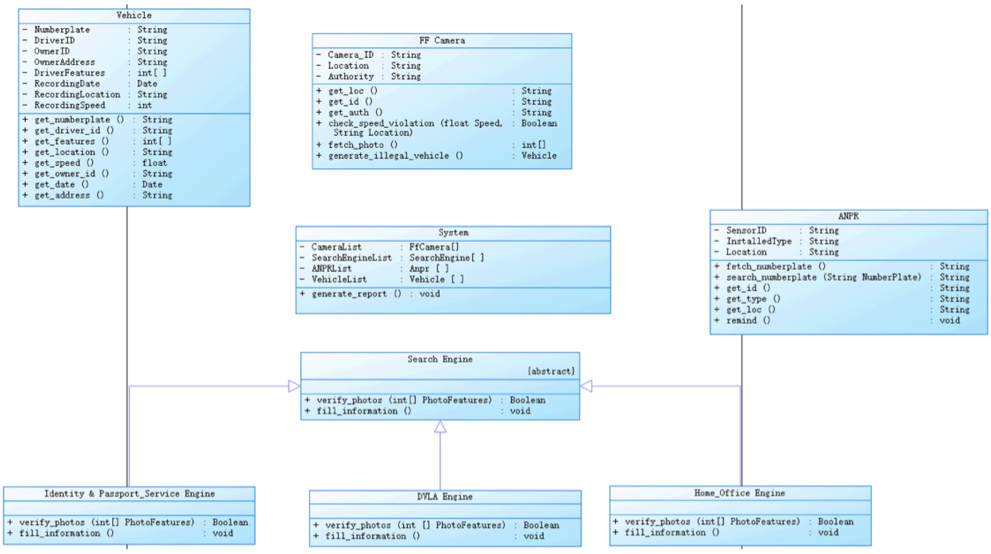
On top of that, the core design of this architecture is to make the proxy server an agency for all the online requests and responses between servers. The proxy server is going to hide both the IP address from the server that sends the request and the server that receives the data. So, anything appears online, even if it is just a request, it is encrypted. This is to protect the system from a backend hack.

The CDN network refers to Content Delivery Network. It is a prevailing technology that helps to share the burden of the server and make access to the information of the system fast from across the country. Meanwhile, it is an effective way of protecting the server from the well-known DDos attack.

If something bad happens to the servers. First, every server is equipped with an inbuilt security system. And second, we have separate git controls for every single server and database. They back up the server on a regular basis, just in case the server is damaged, we can easily recover the server to a previous status at any time point.

3.5 Class Diagram

This diagram illustrate the class models of the DIAMOND system.


Vehicle
- Numberplate : String
- DriverID : String
- OwnerID : String
- OwnerAddress : String
- DriverFeatures : int[ ]
- RecordingDate : Date
- RecordingLocation : String
- RecordingSpeed : int
+ get_numberplate () : String
+ get_driver_id () : String
+ get_features () : int[ ]
+ get_location () : String
+ get_speed () : float
+ get_owner_id () : String
+ get_date () : Date
+ get_address () : String
FF Camera
- Camera_ID : String
- Location : String
- Authority : String
+ get_loc () : String
+ get_id () : String
+ get_auth () : String
+ check_speed_violation (float Speed, String Location) : Boolean
+ fetch_photo () : int[]
+ generate_illegal_vehicle () : Vehicle
System
- CameraList : FfCamera[]
- SearchEngineList : SearchEngine[ ]
- ANPRList : Anpr [ ]
- VehicleList : Vehicle [ ]
+ generate_report () : void
ANPR
- SensorID : String
- InstalledType : String
- Location : String
+ fetch_numberplate () : String
+ search_numberplate (String NumberPlate) : String
+ get_id () : String
+ get_type () : String
+ get_loc () : String
+ remind () : void
Search Engine
{abstract}
+ verify_photos (int[] PhotoFeatures) : Boolean
+ fill_information () : void
Identity & Passport_Service Engine
+ verify_photos (int[] PhotoFeatures) : Boolean
+ fill_information () : void
DVLA Engine
+ verify_photos (int [] PhotoFeatures) : Boolean
+ fill_information () : void
Home_Office Engine
+ verify_photos (int[] PhotoFeatures) : Boolean
+ fill_information () : void

Graph3.5 - Class diagram

3.5.1 Violation Class

This class is designed for recording information of cars which have violation behaviours, their driver and their owners.

Attributes Name	Data Type	Meaning
Numberplate	String	The numberplate of the violation car
DriverID	String	ID of drivers when being recorded
OwnerID	String	ID of the owner of the violation car
Owner Address	String	Address of the owner of the violation car
DriverFeatures	Int [ ]	The face features data
RecordingDate	Date	Date when it’ s being recorded
RecordingLocation	String	Location where it’ s being recorded
RecordingSpeed	Int	How fast it is when being recorded

Table3.5.1 - Attributes of violation class

Function Name	Application
get_numberplate()	Get values of corresponding private attribute
get_driver_id()	Get values of corresponding private attribute
get_features()	Get values of corresponding private attribute
get_location()	Get values of corresponding private attribute
get_speed()	Get values of corresponding private attribute
get_owner_id()	Get values of corresponding private attribute
get_date()	Get values of corresponding private attribute
get_address()	Get values of corresponding private attribute

Table3.5.2 - Public functions of violation class

3.5.2 FF Camera Class

This class is designed for getting the information of cameras and driving cameras.

Attributes Name	Data Type	Meaning
Camera_ID	String	ID of camera for identificaiton
Location	String	Where the camera is installed
Authority	String	The authority this camera belongs to

Table3.5.3 - Attributes of FF Camera class

Function Name	Application
get_loc()	Get values of corresponding private attribute
get_id()	Get values of corresponding private attribute
get_auth()	Get values of corresponding private attribute
check_speed_violation(float Speed, String Location)	Check whether the recorded car violate the speed rule.
fetch_photo()	Take a photo for drivers and get the face features
Generate_illegal_vehicle()	Create an instance for the violation car and fill some information of it

Table3.5.4 - Public functions of FF Camera class

3.5.3 System Class

This class is designed to create a core of this system which can drive other cooperation parts.

Attributes Name	Data Type	Meaning
CameraList	FfCamera[ ]	All camera instances in this system
SearchEngineList	SearchEngine[ ]	3 search engine instance list
ANPRList	Anpr[ ]	All ANPR devices in this system
VehicleList	Vehicle [ ]	All violation cars needed to be search in database

Table3.5.5 - Attributes of System class

Function Name	Application
generate_report()	Generate report for drivers of violation cars to remind them of violation-information.

Table3.5.6 - Public functions of System class

3.5.4 ANPR Class

This class is designed for getting information of ANPR devices and using them.

Attributes Name	Data Type	Meaning
SensorID	String	The unique identity of each ANPR devices
InstalledType	String	The type of ANPR devices
Location	String	Where the devices install

Table3.5.7 - Attributes of ANPR class

Function Name	Application
fetch_numberplate()	Get numberplates of cars from view.
Search_numberplate()	Search detected numberplate of cars in the violation recording
get_id()	Get values of corresponding private attribute
get_type()	Get values of corresponding private attribute
get_loc()	Get values of corresponding private attribute
remind()	Show notification if the detected car has violation behaviours

Table3.5.8 - Public functions of ANPR class

3.5.5 Search Engine Class

This class is designed to search drivers' information in different databases.

Function Name	Application
verify_photos()	Search face data in the database to verify drivers or owners.
Fill_information()	According to the photos, find the information of persons and fill the attributes of vehicle instance.

Table3.5.9 - Public functions of Search Engine class

4 Performance

4.1 Scorecard

This section gives a scorecard that is used to measure the performance of the system, it is also implemented in the dashboard.

View	Measure	Expected	Actual	Deviation	Plan of Action
Customer	Number of Policy Breaches by Employees	1	5	4	If an employee attempts something above their access such as modifying/deleting a record, this must be logged and notified to the dashboard, so the breach can be investigated further.
Local Authority Accounts Online	343	343	0	To determine how many accounts of the web application are online for accessing reports for the different local authorities. This is to monitor who is accessing the system, and that if the number is unexpectedly high, it could suggest unauthorized access by intruders.	
Number of ANPR hits Detected (Hourly)	150	150	0	This is to demonstrate that the DIAMOND system is working and connected with the ANPR devices. This should be correct. Low values that may arise from this could demonstrate issues with the ANPR sensors or the connectivity of DIAMOND. This can be tested using dedicated testing ANPR sensors, with a dedicated record in the database to test it against.	
Number of Speeding Violations Detected (Hourly)	1000	1000	0	This value is used to ensure that other aspects of the system work and ensure that the system and Speed Cameras are functional. Comparing this value to the number of reports generated will demonstrate if the system is working as expected.	
Number of Reports Sent (Hourly)	1000	1000	0	This should be the same value as the number of speeding violations that are detected by the Speed Cameras. If there are any discrepancies, it may suggest that the DIAMOND system is not working as expected, and investigation needs to take place on why, so that the problem can be resolved.	

Reports Confirmed Identified Driver (Monthly)	80%	60%	-20%	The purpose of this system is to ensure drivers who were driving the car cannot claim to not be driving. The success of this should be measured to ensure the system is fulfilling its purpose by indicating the percentage rate of cases drivers have been identified from the photo.	
Improvement	Incorrectly Identified Drivers	5	50	45	Reports sent to drivers should be utmost accurate to prosecute and find fines. If it is reported that drivers have been misidentified by the DIAMOND System, the numbers should be displayed. There will be some discrepancies, but if there are too many it may indicate an issue with the facial recognition program.
Employee Training	100%	60%	-40%	All employees should be trained regarding the digital security implications of the system. Employees should partake in training sessions as soon as possible. Without employees being trained it can make the system much more vulnerable to attacks, especially with social engineering.	
Weekly System Complaints/Feedback	1	5	5	Complaints can be submitted by the users and customers of the DIAMOND service. Taking complaints allow for on the spot feedback for the live system, with some that could point to fatal flaws, security breaches, or reporting employees for their malicious activity.	
System/Firmware Updates	2	5	3	This should be displayed to notify stakeholders when system is due to be down for maintenance, and how long it will be down.	
Operational	Server Load	40%	80%	40%	The network traffic towards the server needs to be monitored to ensure that there are only specific devices communicating with it and that no malicious denial of service attacks is aimed towards the server in question.
Frequency of system offline	1	4	3	Add backup servers and perform load balancing to distribute the traffic.	
Server Status	5	5	0	Servers should be always up and running for the DIAMOND system to be functional. If server status is not normal an investigation should be started to test the connection towards the server, and a physical	

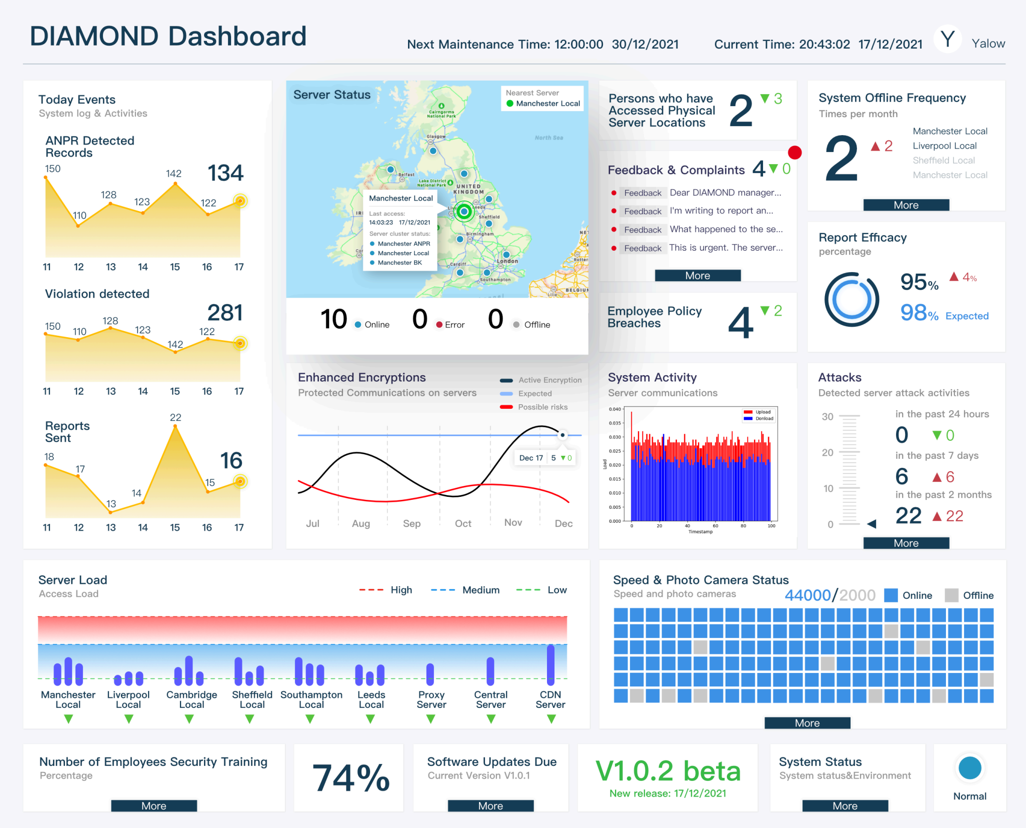
				inspection to assess if the server is up and running.	
ANPR/Camera Status	48000	46000	-2000	ANPR sensors and Speed Cameras should be operational to capture data. It is expected that several cameras/ANPR sensors may be down due to maintenance or unexpected issues.	
Detected Cyber Attacks (Daily)	5	1	-4	In the unlikely case that an individual attempts to attack the system. The security software should notify the dashboard of this, where the attack is being aimed at and where it is potentially coming from.	
Server Firewall Status	5	5	0	To demonstrate that Firewalls are active and working. If this is not the case, extra precautions must take place to ensure that the system is not prone to attack with the firewall to be fixed and become operational as soon as possible.	
Backup Status	5	5	0	This should indicate that all backups are ready, so that if the main system goes down, the backup can take control.	
Database Leaks and Compromises	0	20	20	If unauthorised queries have been run on the Databases on the servers, this should be detected, as it could be a sign of an attacker gaining access to a person’s extremely private information. If this occurs it should be investigated immediately and prevented with blocking access or the worst case scenario of shutting the server down and isolating it.	
Financial and Risk Management	Operation and Maintenance Costs (Monthly)	£120,000	£160,000	£30,000	This provides an idea of the cost of the DIAMOND system, to share with stakeholders, so that cost benefit analysis can be done. Also displaying maintenance costs to allow for budgets to be made that ensure the system is running.
Daily Physical Access to Server	4	4	0	To show how many personnel have had physical access to the server. If any problems with the server occurs, persons who have had physical access may be linked to the issue.	
Maintenance Time/Date	N/A	N/A	N/A	This should notify stakeholders when the system is due an update, so that they are aware when it is next down so preparations can be made to ensure that the maintenance	

				occurs smoothly.	

Table4.1.1 - Scorecard for performance audit

4.2 Dashboard

The figure below demonstrates the conceptual design of the dashboard for the Diamond system using the Scorecard above. Discussions were made between our team and different teams in our cohort on what we plan to display on our dashboard, and what information would be useful.



Graph4.2.1 - Governance Dashboard

4.3 Safety

When discussing the safety conformance of the DIAMOND system, there are two problems that need to be considered: whether the system meets the requirements and what kind of problems the system may face.

The requirements for system functions were defined specific and appropriate. The plan we have for measuring whether the system fulfills the requirements accurately and appropriately is as follows:

4.3.1 Feasibility of requirement realization

> **Review:** whether the code accurately fulfill system requirements.

Possible problems: Inadequate structural or functional design of code may result in structural problems or missing functionality in parts of the system.

Test method: Test the code according to the system function. Use the test data set to test the code to check the integrity and accuracy of the functionality implemented.

Audit: System testing personnel should test the system's code to see if the system meets all requirements. If any vulnerabilities are found, contact system maintenance personnel to optimize the system.

4.3.2 Function realization degree of the system

> **Review:** Whether the driver's facial features can be accurately extracted.

Possible problems: the system may not be able to accurately extract the facial features of drivers because of the defects of extraction algorithm, which may lead to failure to find valid results in the subsequent photo information comparison process.

Test method: Test the system with a specified test dataset, observe the comparison result between the features extracted from the dataset and the original data.

Audit: The government should employ professional image processing researchers to test and improve the performance of the image feature algorithm.

> **Review:** Whether the sent requests can be received by the DVLA/IPS/Home Office server.

Possible problems: A large traffic flow in a special period may lead to an increase in the number of traffic offence, it can cause the system to generate massive requests in a short period of time, which will cause great stress to the server.

Test method: Make a stress test to the system, input a lot of violation records into the system and find the maximum number of information comparison requests that the system can process simultaneously.

Audit: The system maintenance personnel hired by government should adjust the system function according to the results of the pressure test. If necessary, expand the capacity of central processing server and limit the generating speed of requests based on the total number of violation records.

4.3.3 Consistency between system structure and required functions

The following paragraphs analyzes the risks that the system may encounter and how the system architecture is designed to against them.

> **Review:** Whether the system has the ability to deal with a large number of violation records in a short time.

Risk: There are so many violation records from speed cameras that the Local Server cannot process them for a while.

Solution： The local offline backup server which contains in the system will store all violation records and it avoid server overloads or obliterated data.

Test method: Make a stress test to the system, input a lot of violation records into the system and find the maximum number of records that the system can process in a short time.

Audit: Database maintenance personnel employed by local office should periodically check the integrity of the data and ensure that the database will discard expired data according to the expiration date of the data.

> **Review:** Whether the system has the ability to process a large number of information matching tasks and generate corresponding reports in a short time.

Risk: The central processing server is not only responsible for the processing of information, but also for the generation of reports. Heavy tasks can easily lead to server overload and even affect system efficiency.

Solution: A proxy server was set to share tasks with central processing server. The central processing server sends tasks to the proxy server, which protects the security of the central processing server.

Test method: Make a stress test to the system, input a lot of violation records into the system and find the maximum number of records that the system can process in a short time.

Audit: Server maintainers hired by the government should optimize the system based on the stress test results. The information processing capability of the system should be enhanced according to the difference between the actual traffic and the maximum traffic and enhancing the capacity of the proxy server is also considered.

4.3.4 Whether the system has technical defects or application defects

System technical defects and application may be caused by the defects of system algorithm or system structure. If the tester does not thoroughly investigate the system process, the defects are difficult to find.

> **Review:** Whether the system has technical defects.

Risk 1: Vulnerabilities caused by technical defects may result in mismatches of photographic information and even leads to information hoarding/ information loss. In the worst cases, it can cause a system crash.

Test method 1: Test cases should be divided according to functions, each function of the system must be tested separately, and the test results should be recorded. This process requires the cooperation of multiple areas of staff to find the system's technical vulnerabilities.

Audit 1: Testers hired by the government should conduct a comprehensive test of the system by testing and optimizing multiple nodes of the system and analyzing system functions with different test data. After finding the technical defects, the corresponding technicists should figure out a corresponding scheme and repair the defects.

Risk 2 : There may be some problems in the implementation of system functions, which will affect the user experience.

Test method 2: Test the system from the application layer. Test the system application according to the type of system users. Select a certain number of real users of the system, collect their feedbacks, and update or repair the system after analyzing the feedbacks.

Audit 2: According to different types of system users, the user experience of police, DVLA staff and local Authority staff was collected respectively. Investigate whether the information received by different users is consistently and correctly. If there is information discrepancy, the system should be investigated according to the information source to find out the defects.

4.3.5 Whether the system can perfectly solve the possible security problems

> **Review:** Whether the system has a backup plan for the failure of some nodes in the system.

Test method: Manually stop some components of the system, check whether the rest of the system is affected, and observe whether different users of the system can still use the system normally and still get the correct information for the system.

Audit: The system testers and system maintenance personnel employed by the government should evaluate the system stability based on the results of breakpoint test and check whether the backup server and backup database in the system can maintain the normal operation of the system. If the system fails due to a partial structural collapse, the staff should submit a new backup plan and make structural improvements to the system.

> **Review:** Whether the system can cope with malicious attacks from external environment.

Test method: Let supervisory personnel simulate malicious attacks to test whether the system can protect information under attack and ensure the integrity and accuracy of information.

Audit: Government need to check whether the DIAMOND systems have the ability to protect information from illegal modified or leaked. Grade the system based on its performance and optimize the system structure according to system weaknesses.

4.4 Reliability

4.4.1 System operations

> **Review**: What are the system permissions that different accounts have?

Test: Log in to the accounts of users with different permissions, and check whether their access permissions are less than or more than the declared permissions.

Audit: The accounts permissions should be checked at least once before the formal use, and then the inspection frequency can be reduced. However, if someone reports abnormal permissions, it needs to be checked in time.

> **Review**: Does the system have the functions it should have?

Test: Check whether the operations in the system are consistent with the required operations.

Audit: The functions should be checked at least once before formal use, and then the inspection frequency can be reduced. If new functions are added, it needs to be rechecked.

> **Review:** Whether each operation in the system is consistent with expectations?

Test: Perform each operation in the system at least once.

Audit: Check these operations regularly. Collect feedback, if any operation needs to be improved, it should be modified.

4.4.2 System load and capacity

> **Review:** What is the carrying capacity of the system?

Test: Increase the number of online users of the system.

Audit: Record the maximum capacity of the system and avoid reaching the maximum capacity during daily use. If the daily capacity is close to the highest, expansion needs to be considered.

> **Review:** What is the response time of the system under high load?

Test: 1. Operate the system when there is large number of online users. 2. Increase the frequency of camera taking pictures of the vehicle, then operate the system when a large amount of data is being transmitted.

Audit: Record the response time / transmission speed under high load. Ensure that the system can operate normally even under high load.

4.4.3 System crashes

> **Review:** If the system crashes, what is the recovery time and loss?

Test: Simulate the system crash, and then restore the system.

Audit: Record the time required for system recovery If this time is not acceptable, the scheme for restoring the system should be improved.

4.4.4 Connection reliability

> **Review:** Is the connection between each part of the system fast and reliable?

Test: 1. Take photos with hardware (e.g., ANPR) and upload them to the system, record the upload time. 2. Carry out data transmission between each part of the system and record the transmission time.

Audit: Regularly check the connection of all parts of the system If the connection is unstable or not fast enough, it needs to be repaired in time.

4.4.5 Report accuracy

> **Review:** What is the accuracy of the report?

Test: 1. Check whether the generated report is accurate (such as face and personal information, vehicle information, etc.) 2. Check whether the report is consistent with the database

Audit: Check the report regularly and compare the information in the report in many aspects (such as the database of each department). In case of any abnormality, it needs to be checked and repaired in time (such as API error)

4.5 Availability

Definition: The percentage of time that a system remains operational when required.

Calculation: Divide uptime by the sum of uptime and downtime.

$$Availability = \frac{uptime}{uptime + downtime}$$

Therefore, the main aspect when reviewing the availability is the downtime.

4.5.1 Stuff permissions

> **Review**: Do the permissions of different employees to operate the system meet the requirements?

Test Plan: Check if the permissions of each employee meet the requirements.

Audit: Regularly check if the test has been done and the effectiveness of the test: if it can really reduce the possibility of system downtime due to employee misoperation.

4.5.2 System bugs

> **Review**: Does the code have bugs that will lead to a downtime of the system?

Test Plan: Check each module, function and operate each performance to see if there are some bugs. Then fix them before the releasing.

Audit: Regularly check if the test has been done and the effectiveness of the test: if it can really reduce the possibility of system downtime due to system bugs

4.5.3 Software availability

> **Review**: What’ s the availability of the software, such as servers and databases?

Test Plan: Simulate the real-world high concurrency: such as large volume of image data, large number of online users, to test the maximum capacity, connections between servers and data backup process. Then optimize the software architecture to improve the availability.

Audit: Regularly check if the software availability test has been done and the effectiveness: if it can really measure and improve the software availability. Then continually optimize the architecture of the servers and databases, such as loading balancing, sub-library and sub-table.

4.5.4 Hardware availability

> **Review**: What’ s the availability of the hardware, such as ANPR sensors and speed camera?

Test Plan: Periodically record the aging degree of the ANPR sensors and speed camera. Record the average aging time. Develop a timetable for equipment repair and replacement. Because ANPR sensors and speed camera need to be 24/7, a reasonable maintenance plan can better ensure their availability.

Audit: Regularly check if the equipment repair and replacement has been done and the effectiveness: if it can really improve the hardware availability. Then continually optimize the timetable for equipment repair and replacement.

4.5.5 Maintenance frequency

> **Review**: What’ s the frequency of a planned downtime for system maintenance?

Test Plan: Monitor system traffic in real time and select the low peak for maintenance, such as midnight. Meanwhile, develop a timetable of planned downtime for system maintenance: the frequency of maintenance and duration of each maintenance.

Audit: Regularly check if the maintenance has been done and the effectiveness: if it can really improve the overall availability of the system. Then continually optimize the timetable for system maintenance.

4.5.6 Recover ability

> **Review**: If the system (server, network, database) fails, what’ s the ability to recover?

Test Plan: Simulate different types of system crash, such as damage of database, network and server, then test the mean recovery time and optimize the damage recovery mechanism.

Audit: Regularly check if the recover ability test has been done and the effectiveness: if it can really improve the recover ability. Then continually optimize the mechanism for damage recovery.

4.6 Resilience

> **Review:** This part is to introduce how this system can recover from a destroying event or a malicious accident.

Test Plan:

1 External damage

a) Camera Damage (including electricity problems)

Test Content: Cut off the power of some cameras or block communication via network to see whether the system will notice that some of the cameras are broken and inform responsible people.

Solution: When cameras cannot send all the information (include driver features, locations, time, numberplate, speed), the system will send a notification to inform that which camera needs to be repaired. Each camera has its unique camera_ID, location and authority information so it is easy to be found.

b) Server Damage (Including electricity problems)

Test Content: Cut off the power of some Server to simulate damage in order to see whether the system can continue working.

Solution: When one server is broken down, the system will automatically choose backup server to continue working.

2 Internal damage

a) System Breakdown

Test Content: Cut off the power of central computer and restart it to see whether the system will recover.

Solution: When the system restarts, it will set the server according to its git repository. If it has the unfinished process, it will recover the process according to the log.

b) Data Lost

Test Content: Artificially delete some schemas of database to check whether the system will recover the original database.

Solution: When the system starts, it will check the database according to the system log, if it finds some error, it will recover the database by the latest version from the backup server.

Audit: For each test, check whether this system can recover and how much time it takes to recover. Check whether this system can ensure that data can be stored perfectly and can recover even experiencing damage. What’ s more, evaluate the ability of repairing the server or database.

4.7 Security

1 Possible hardware security problems

> **Review:** Hardware aging and damage test caused by the natural environment.

Test: simulate some extreme weather in the natural environment, evaluate the service life of the hardware in these scenarios, and formulate the hardware installation plan and protection plan.

Audit: The government should regularly send special inspectors to check and maintain the equipment, to know whether the hardware equipment after the protection plan, has played a role in extending the service life.

> **Review:** The hardware device is damaged or is damaged in an emergency.

Test: In the case of simulated sabotage, the hardware can give instructions to the attached hardware to keep them working and send reports to the staff or police.

Audit: Assess whether there is a timely report of equipment damage and whether nearby equipment can help.

2 Possible software security problems

> **Review:** Because face information and data are very valuable, hackers may attack system software or steal system information.

Test: The government can employ some professional technical personnel to simulate hacker attack on the system. For example, adding a large amount of information to the system in a short period of time will exceed the system load and make the system crash. The system needs to make a response plan according to the potential attack.

Audit: Real-time monitoring of the operating status of the system. If non-structural abnormalities are found in the system by chance, it is possible to face attacks. Deal with them according to solutions designed by professional and technical personnel, such as calling idle servers to deal with excessive load, and finally adjust and update solutions according to the results.

> **Review:** Because the system is very large and involves a lot of users and hardware devices, there may be mistakes and deficiencies in the overall architecture design when the system is put into use initially.

Test: Before the actual use, this potential error is difficult to avoid, therefore, can take the way of gradual installation to correct.

Audit: With the expansion of the use of the system, some problems that are not reflected at the beginning will gradually manifest. Then, the problems can be corrected according to the previous problems in the subsequent installation and use.

> **Review:** Whether the sent requests can be received by the DVLA/IPS/Home Office server. A large traffic flow in a special period may lead to an increase in the number of traffic offence, it can cause the system to generate massive requests in a short period of time, which will cause great stress to the server and even caused the system to crash.

Test: Make a stress test to the system, input a lot of violation records into the system and find the maximum number of information comparison requests that the system can process simultaneously.

Audit: The system maintenance personnel hired by government should adjust the system function according to the results of the pressure test. If necessary, expand the capacity of central processing server and limit the generating speed of requests based on the total number of violation records.

> **Review:** The central processing server is not only responsible for the processing of information, but also for the generation of reports. Heavy tasks can easily lead to server overload and even affect system efficiency. Therefore, we need to evaluate whether the system can process many information matching tasks and generate corresponding reports in a short time.

Test: A proxy server was set to share tasks with central processing server. The central processing server sends tasks to the proxy server, which protects the security of the central processing server. Moreover, we can make a stress test to the system, input a lot of violation records into the system and find the maximum number of records that the system can process in a short time.

Audit: Server maintainers hired by the government should optimize the system based on the stress test results. The information processing capability of the system should be enhanced according to the difference between the actual traffic and the maximum traffic, and enhancing the capacity of the proxy server is also considered.

3 Possible system users' security problems

> **Review:** Although the system has been automated in many places, it still cannot avoid the need for staff to operate, and manual operation will certainly make mistakes to a certain extent, such as covering up the reports of some speeding drivers, or not timely enough or wrong people when submitting personnel information.

Test: First, the government or police should train relevant staff to ensure that they will not be bribed. Meanwhile, any steps involving submission and review should be supervised by two or more staff.

Audit: Every quarter or every year, special staff should be sent to carry out spot checks to ensure the efficiency and work of staff.

5 People-device symbiosis

Integration for mitigation of security threats from human factors

1. Creating hierarchy of access to mitigate security threat via employee misconduct / error - Insider Threat
2. Double checking revoking access for ex-employees to mitigate security threat via the accidental or unintentional Insider
3. Run assessment checks to flag out vehicles with false number plates.
4. Setup measures to identify drivers wearing face masks to conceal their identity (not face coverings).

Getting the system accepted by its users, to make sure the users are accepting the system, the following measures can be taken:

1. Educating the users about the system which will enable them to see for themselves the advantages the system will bring.

2. Creating a series of tutorials outlining how the system will safeguard the users in adverse situations
3. Organize seminar / events where the users can clarify all doubts / queries they have about the system and its impact

Evaluating risks:

1. Performing daily EOD access audit to ensure that resources/infrastructure had no unauthorized access.
2. Generating reports outlining the measures for false number plates and identity concealing incidents.

Reporting risks and incidents:

1. Since the users will be educated about the functioning of the DIAMOND system, they will be more inclined towards reporting incidents because of the impact it has on them.