GROUP ASSIGNMENT

System Design Studio (SP5 2021)

MADE BY: EXT-GROUP-6
Group member: Gitae Bae, Collins Kiplimo, Chi Lok Luk, Shulin Yuan, Haige Yuan, Hanxiao Zhou

DECLARATION OF CONTRIBUTION

Team Member Name: Hanxiao Zhou

I contributed 385 words and 5 diagrams towards this assessment.
I worked on the following sections/questions/elements:
3. Component Decomposition 4. Decomposition Validation

Team Member Name: Collins Kiplimo

I contributed 231 words and 5 diagrams towards this assessment.
I worked on the following sections/questions/elements:
7. Implementation Design

Team Member Name: Haige Yuan

I contributed 1578 words and 2 diagrams towards this assessment.
I worked on the following sections/questions/elements:
5. System Interface 6. Component Interfaces, 9. Deployment Diagram

Team Member Name: Chi Lok Luk

I contributed 701 words and 1 diagrams towards this assessment.
I worked on the following sections/questions/elements:
2. Volatility Lists 8. Designing Boundaries

Team Member Name: Gitae Bae

I contributed 726 words and 2 diagrams towards this assessment.
I worked on the following sections/questions/elements:
1. Domain Model 8.Designing Boundaries

Team Member Name: Shulin Yuan

I contributed 420 words and 4 diagrams towards this assessment.
I worked on the following sections/questions/elements:
5. System Interface 6. Component Interfaces

1. Domain Model

Domain Classes and Attributes

Call is occurred by patient, it contains call ID, call number, call address, call information, call date, call details of patient.

Patient calls 119 and is important in system, it contains patient ID, name, email, Address, telephone number, location, time, details.

Call maker is able to connect to the call taker as soon as possible so time is saved, it contains call maker ID, name.

Call taker documents the call as well as send case details based on the degree of urgency as a result of following decision trees to thse dispatcher through the system as soon as possible so that the dispatcher can choose the right ambulance crew in time, it contains call taker ID, name.

Case is extent of sore, it contains case ID, details.

Category is divided from 1 to 5, it contains category ID, type.

Dispatcher is able to upload any case along with caller address and other details quickly for the ambulance crew so that the ambulance crew can reach the patients in time, it contains dispatcher ID, name.

Ambulance Crews including 2 paramedics use the system to be able to receive patients details as soon as possible so they can be used to save the patient's life as soon as possible, it contains crew ID, name.

System is core of the this problem statement, it contains system ID, type, machine, details.

Paramedics uses the curing management at urgent situations as part of their job and they are also project stakeholders who need to be consulted about the system as they have a lot of knowledge about how the system is used and any changes which are necessary, it contains paramedic ID, name.

Team leader uses the system to be able to see the usage of ambulance crews in order to determine shifts, it contains team leader ID, name.

Procedure is provided to ambulance crews using system, it contains procedure ID.

Information is provided to ambulance crews using system, it contains information ID, content, type, details.

Policy is provided to ambulance crews using system, it contains policy ID.

Computer is one of the system machines, it contains computer ID.

Terminal is one of the system machines, it contains terminal ID.

Headquarter is in control of outcomes about the system as they have good discovery and fastness about how the system is used, it contains headquarter ID, name.

I.T Department uses the system to be able to maintain the system easily through regular testing, improvements and repairing if needed, it contains I.T department ID, name.

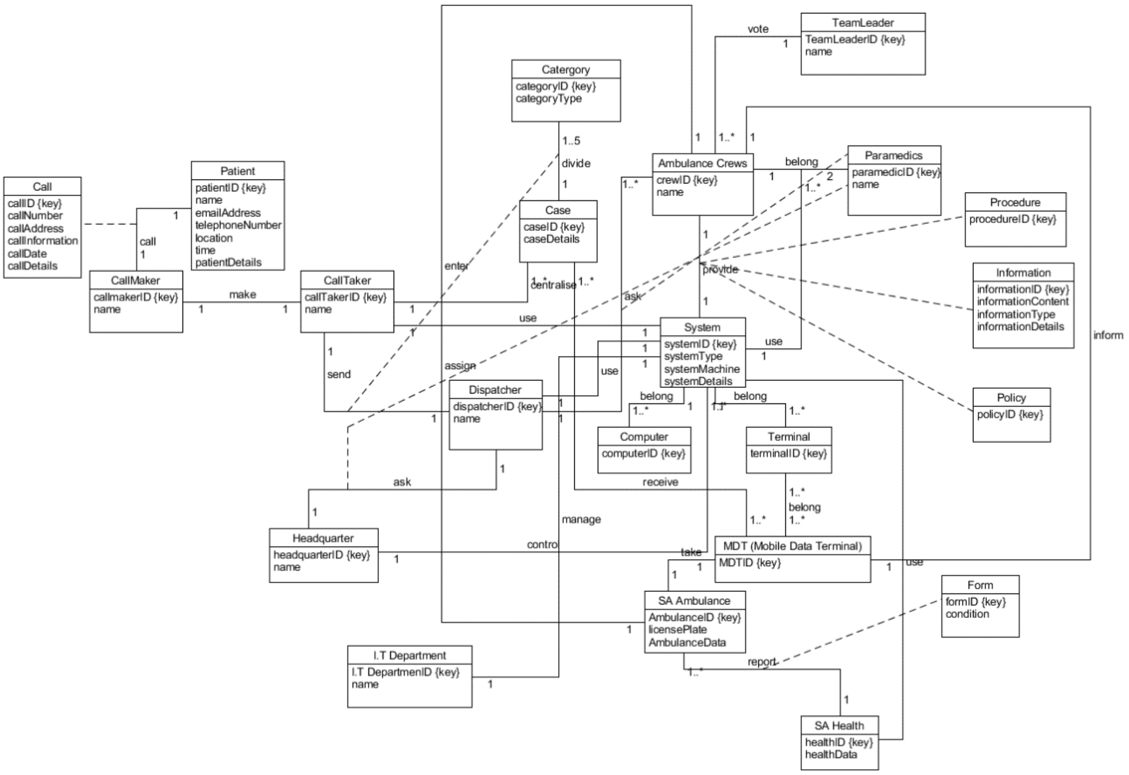
SA Ambulance should be able to use the system to track all data on the system such as number of calls, patients' data, ambulance data, etc, it contains ambulance ID, license plate, data.

SA Health should be able to use the system in order to track data of patients such as: cause of needing emergency service, number of deaths in hospitals, number of patients being affected by different types of diseases, etc, it contains health ID, data.

Form is needed in order to report, it contains form ID, condition.

MDT (Mobile Data Terminal) is belonged to terminal in order to use system, it contains MDTID.

Domain Model Class Diagram



Associations

A patient can call a call maker with call.

A call maker can send a call taker.

A call taker can centralize one or more case using a system and send it to a dispatcher.

A case is divided from 1 to 5 categories using one or more MDT (Mobile Data Terminal).

A headquarter is asked for a dispatcher with paramedics using a system.

A I.T department can manage a system.

One or more ambulance crews are asked for a dispatcher with paramedics and can vote a team leader and an ambulance crew is provided procedure, policy and information to a system and an ambulance crew is belonged 2 paramedics and one or more ambulance crews and paramedics is using a system.

A dispatcher is using a system.

A system belongs one or more computer and terminal.

One or more terminals are belonged one or more MDTs (Mobile Data Terminal).

A SA ambulance takes MDT (Mobile Data Terminal).

A SA Health is reported one or more SA ambulance with form using system.

2. Volatility List

Volatility Lists:

- **Procedures**
- **Storage**
- **Security**
- **Dispatch**
- **Caller**
- **Case**
- **Route**
- **Prank**

Identify Volatilities

I. **Procedures volatility:**
Different hospital (especially personal hospital) may have different procedure. That system do the procedure may get problem delay treatment.

II. **Caller Taker volatility:**
Caller Taker may get wrong decision that may put patient in crisis. so, it should have a function for them to double check the detail.

III. **Storage volatility：**
One accident will create much thing (notification, hoaxes， event detail) in database. The storage will be easy to full. Therefore, upload to cloud maybe a good way to solve it.

IV. **Security volatility:**
Caller will not like the course of disease the other know, so I think three should be a function for just let the patient and the medical personnel check the detail, for the other need to inspection identity or get permit.

V. **Dispatch volatility:**
The dispatcher must consider the urgency of the case, because not all the car in the same area, just like on average, there will be 20 Ambulance Crews within a Headquarters' area, and it may not be enough people when big accident happens like multi-car pileup.

VI. **Caller volatility:**
When the call taker finds the patient, it may not get the local of the patient well, in this time system will access public information from telephone, to get the local of the patient, this can accelerate help the patient.

VII. **Case volatility:**
Some Category case is very dangerous. Although short time ambulance can come but if the caller knows some save way from Call Taker that can Increase patient survival.

VIII. **Route volatility**

Traffic and route problem is not easy to get although the system provide a function can up to driver choose route but if run half distance that something happen (traffic problem suddenly happen) driver cannot be easy to change route or go to this hospital. Therefore, the ambulance need has a live update route function.

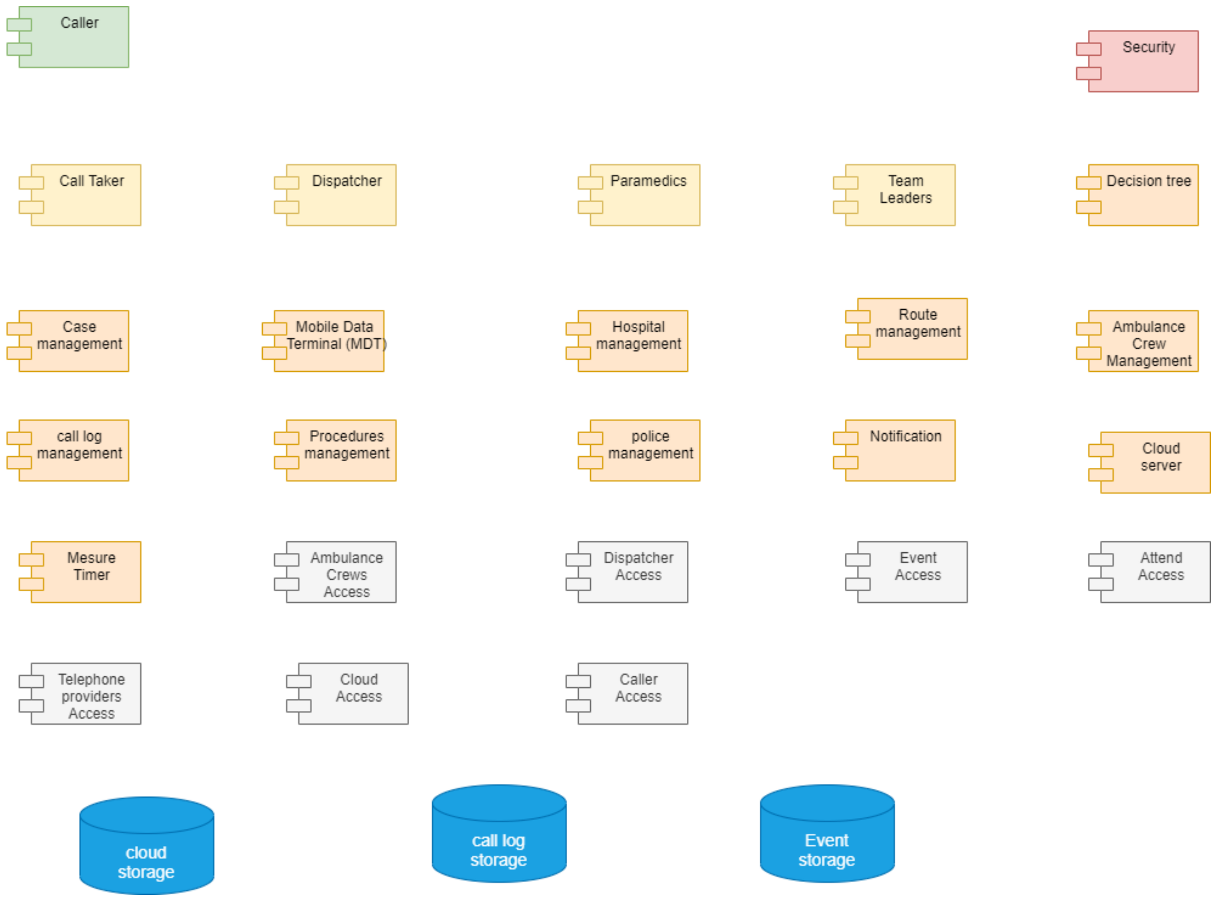
IX. **prank volatility**

Many pranks or pizza call is trouble and the law that will need collect information from the troublemaker. If not get enough evidence. They still will affect work.

X. **Time volatility**

In each case, time is very important, I think that there should be a timer That will go into rang every five minutes to let the rescue to catch the time better, Reduce the risk of the ambulance.

3. Component Decomposition



Component	**Responsibility(ies)**
Caller	Present the user interface of caller
Call Taker	Answer the call. Collect information from caller
Dispatcher	Receive case information Arrange Ambulance Crew to take case
Paramedics	Present interface of paramedics acknowledge and assess the case, drive Ambulance to the location, go to hospital
Security	Caller authentication and authorization
Decision tree	categorization of the case to Category1-5,

	guide and ask Call Taker to double check and make sure what the case is.
Team Leaders	operations during shift reports of Ambulance Crew usage during the shift.
Case management	Arrange case detail. Provide way can help the case solve (like, drink some water or Artificial respiration way)
Mobile Data Terminal (MDT)	receive assigned case from dispatch Tell the Ambulance Crew where the case location is.
Hospital management	Find recent hospital and provide hospital detail.
Route management	Calculated Path and traffic Provide route to the location
Notification	Notice to Paramedics in the Ambulance Crew
Ambulance Crew Management	Arrange Paramedics to take case.
Call log management	Verify the call whether it is a prank
Procedures management	create procedures for the case and send it to hospital
Police management	According to local's police do suggestion (e.g: run Red light request)
Cloud server	integrated into the Triple-Zero Call Taker's application
Caller Access	Record the information (location, time, case) from the caller
Ambulance Crews Access	Retrieve and record Paramedics details
Dispatcher Access	Retrieve and record Dispatcher details
Event Access	Retrieve and record the whole event details
Attend Access	Record Paramedics route and signal details

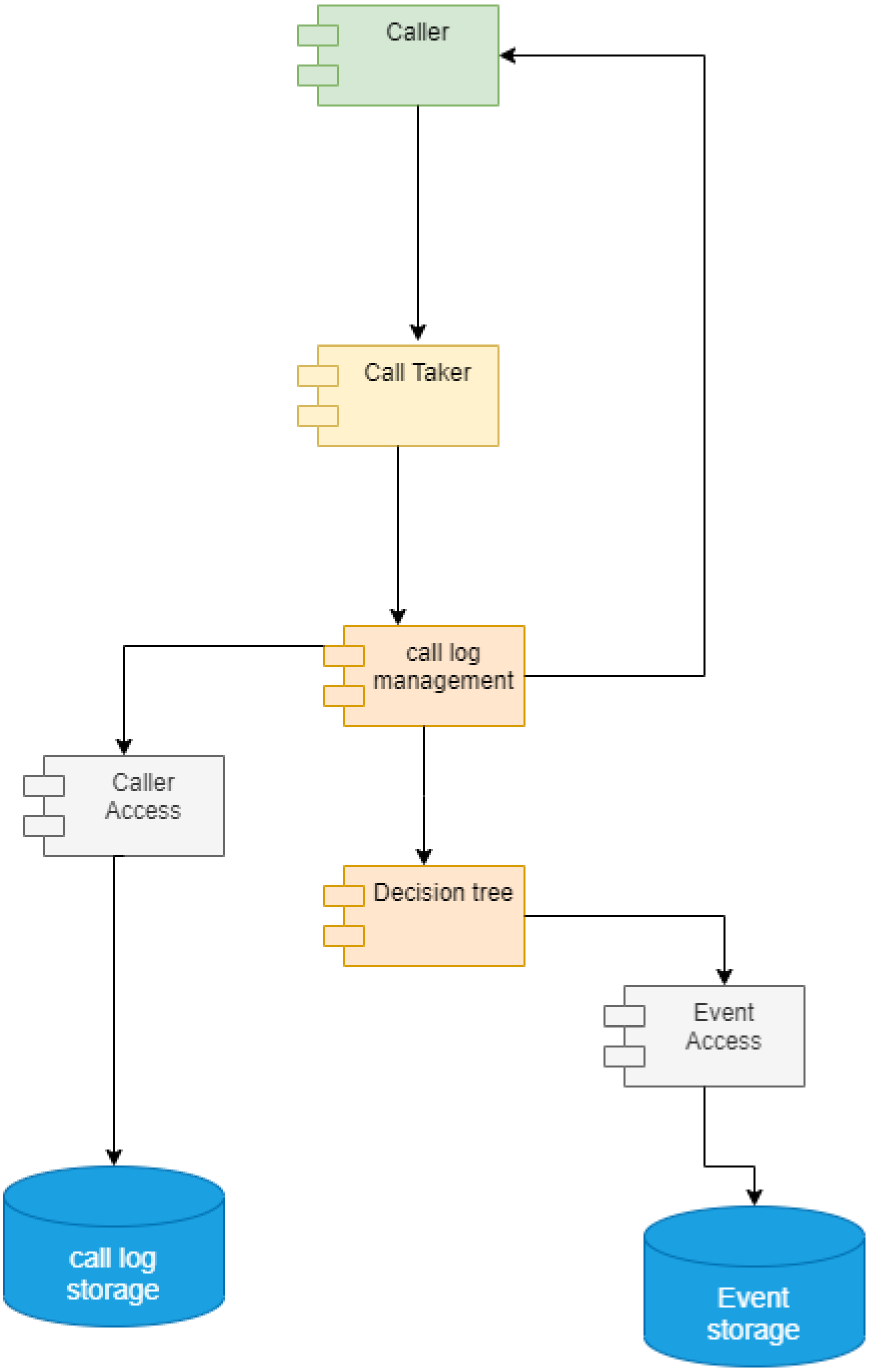
Telephone providers Access	Access to the caller telephone’s providers
Measure Timer	Measure time and remind paramedics
Cloud Access	Access to the external cloud server. Upload data in it

Volatility map

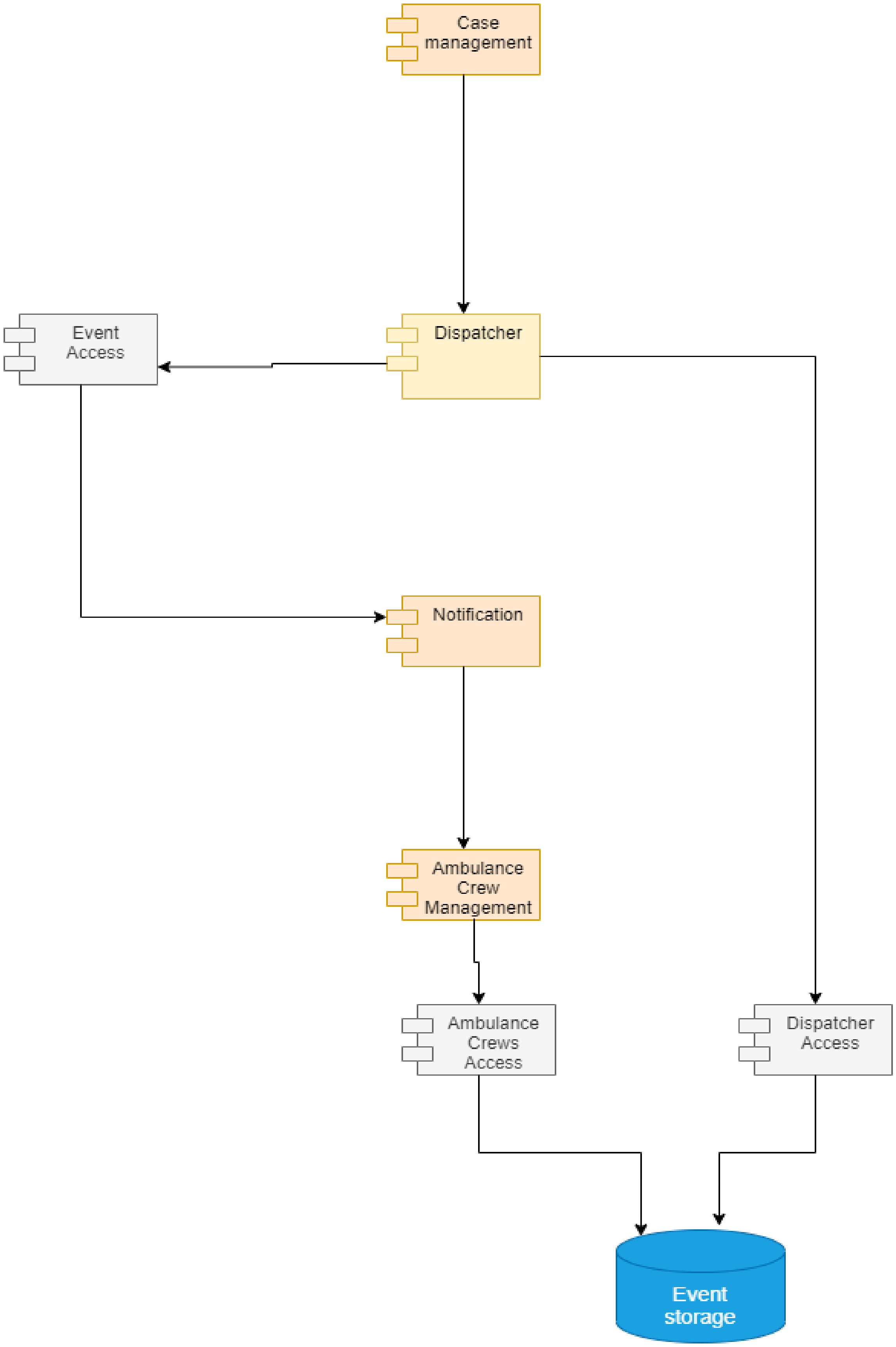
Volatility	Detail	Encapsulated in
Procedures	Solve procedures problem	Procedures management
Storage	Upload event detail to online cloud storage	Cloud Access
Security		Security
Dispatch	Arrange Paramedics and if not enough get call to other crew to get help	Ambulance Crew Management
Caller	According to Telephone providers to get location	Telephone providers Access
Case	Collect case information and provide way to Call taker and Call Taker tell the caller	Case management & Call Taker
Route	Manage route and live update show in the ambulance	Route management
Prank	If get prank call. Call log management will record it	Call log management
Time	Measure time and remind paramedics	Measure timer

4. Decomposition Validation

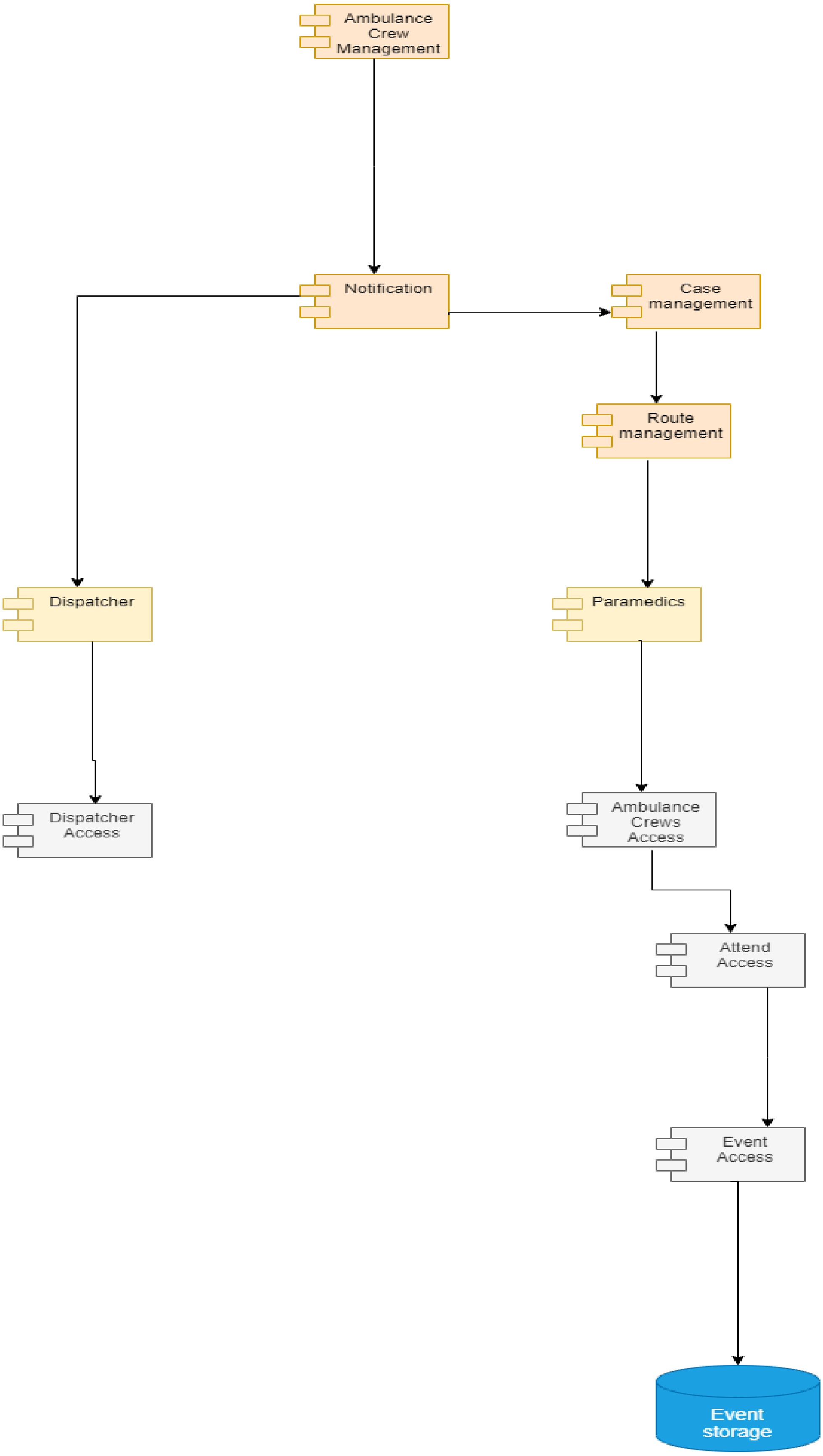
UC01: Receive Emergency Call



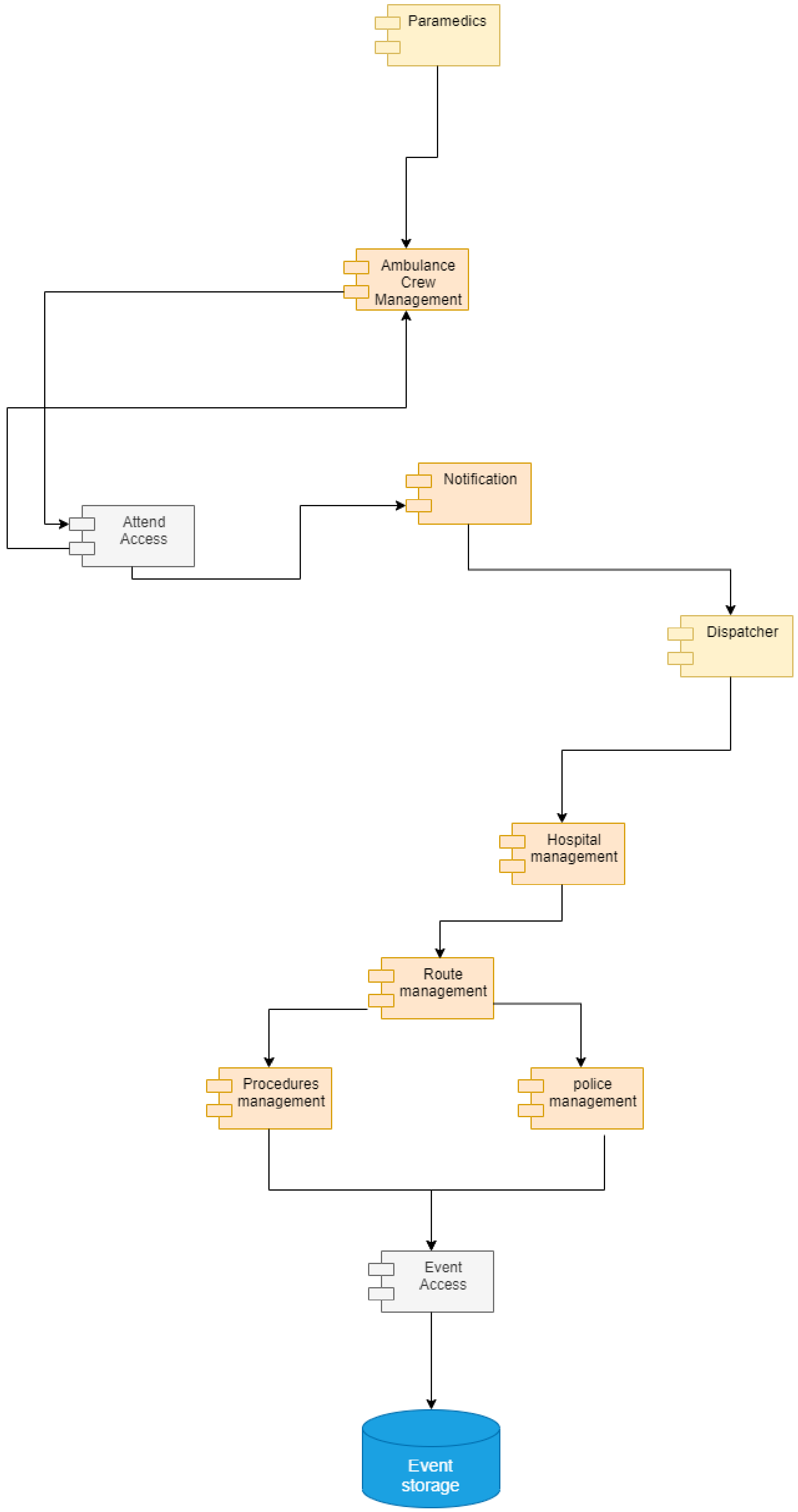
UC02: Dispatch a Case



UC03: Acknowledge and Attend Case

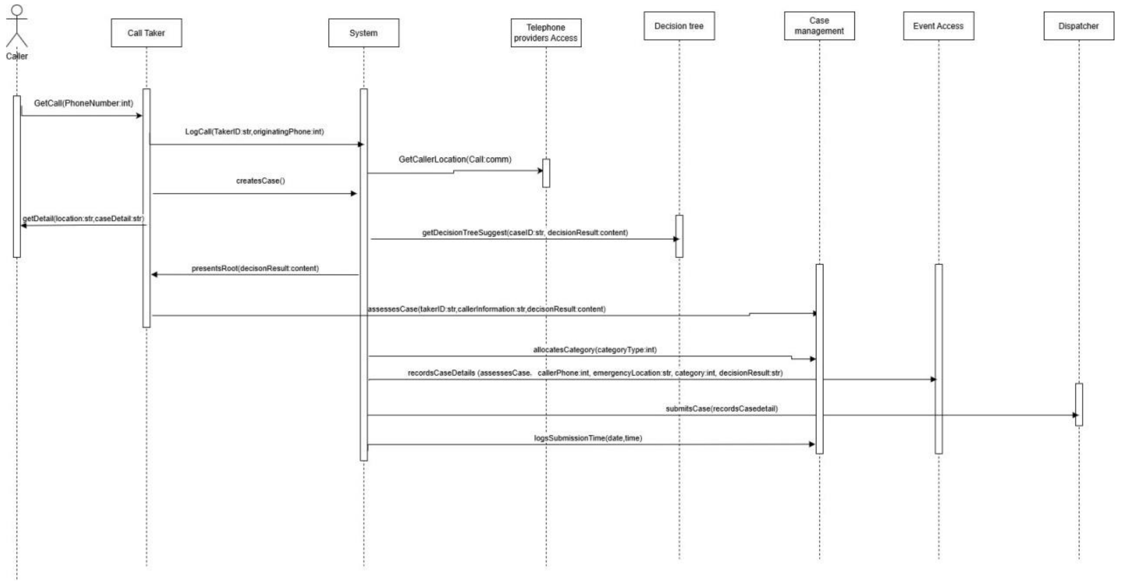


UC04: Assess Case



5. System Interface

UC01: Receive Emergency Call



Caller
Call Taker
System
Telephone providers Access
Decision tree
Case management
Event Access
Dispatcher
GetCall(PhoneNumber:int)
LogCall(TakerID:str,originatingPhone:int)
GetCallerLocation(Call:comm)
createsCase()
getDetail(location:str,caseDetail:str)
getDecisionTreeSuggest(caseID:str, decisionResult:content)
presentsRoot(decisonResult:content)
assessesCase(takerID:str,callerInformation:str,decisonResult:content)
allocatesCategory(categoryType:int)
recordsCaseDetails (assessesCase, callerPhone:int, emergencyLocation:str, category:int, decisionResult:str)
submitsCase(recordsCasedetail)
logsSubmissionTime(date,time)

1) GetCall(PhoneNumber:int)
Caller call the number of Emergency

pre-condition:caller has called to the caller taker

post-condition: caller taker had already obtained the call, and the system had already record the call((TakerID,originatingPhone), and then getCallerLocation(call:comm) operation was triggered.

2) LogCall(TakerID:str,originatingPhone:int)
The System logs the call, the Call Taker, the originating phone number

Preconditions: instanceof callRecording,callTaker,phoneNumber exist

Postconditions: Acall instancewas created getCallerLocationoperationwas triggered

3) GetCallerLocation(Call:comm)
The System logs the approximate caller location

Pre-condition:the call was received by the caller taker.

Post-condition:the system was obtained the accurate location from the telephone provider, and then getDetail(location:str,caseDetail:str) operation was triggered

4) createsCase()
Call Taker creates a case for the call

Preconditions: aninstanceof caseDetailsexist

Postconditions: Acaseinstancewas created getDecisionRoot operationwas triggered

5) getDetail(location:str,caseDetail:str)

Call Taker asks the Caller for details

pre-condition:the system had already gained the location from the phone provider

post-condition: the caller had already got the detail from the caller taker, and then getDecisionTreeSuggest(caseID:str, decisionResult:content) operation was triggered

6) presentsRoot(decisonResult:content)
Call Taker assesses the case based on the information provided by the decision tree

pre-condition:the system was got the suggestions from the decision tree

post-condition:the system had already displayed the root to the caller taker, and then assessesCase(takerID:str,callerInformation:str,decisonResult:content) operation was triggered.

7) getDecisionTreeSuggest(caseID:str, decisionResult:content)
The System presents the root of the decision tree to the Call Taker

Pre-condition:the caller was received the detail

Post-condition:the suggestions(decision tree) had already gained by the system, and then presentsRoot(decisonResult:content) operation was triggered.

8) assessesCase(takerID:str,callerInformation:str,decisonResult:content)
Call Taker assesses the case based on the information provided by the Caller and the decision tree

pre-condition: the system had already displayed the root to the caller taker

post-condition: the case had already accessed by the case management, and then allocatesCategory(categoryType:int) operation was triggered

9) allocatesCategory(categoryType:int)
The System allocates a category to the case

pre-condition: the case had already accessed by the case management

post-condition:the system had gained the category type from the case management, the system was recorded the case details in the event access,and then submitsCase(recordsCasedetail) operation was triggered.

10) recordsCaseDetails (assessesCase， callerPhone:int, emergencyLocation:str, category:int, decisionResult:str)

Records the case details (call, emergency location, category, decision tree)

pre-condition: the case had been created

post-condition: Submit after record is completed

11) submitsCase(recordsCasedetail)
submits the case to an appropriate Dispatcher

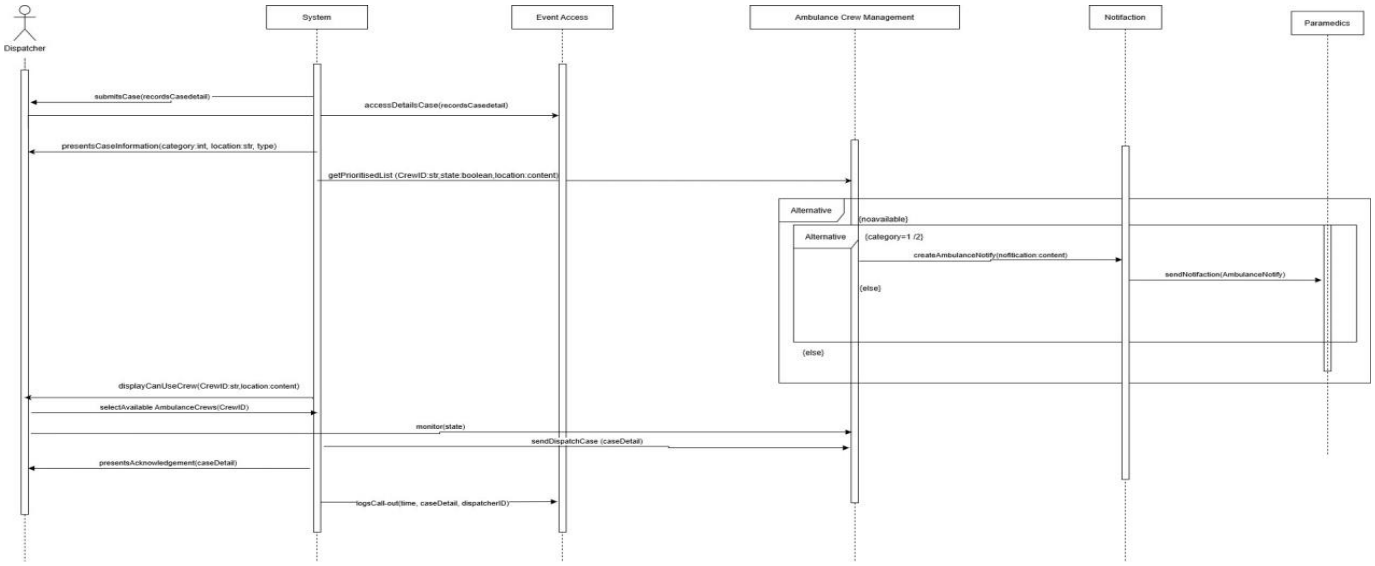
pre-condition: the details of the case was recorded by the system

post-condition:the dispatcher had already received the case, and the system was recorded the submission time to the case management, and then accessDetailsCase(recordsCasedetail) operation was triggered.

12) logsSubmissionTime(date,time)
logs the submission time

pre-condition: submission has been finished

UC02: Dispatch a Case


Dispatcher
System
Event Access
Ambulance Crew Management
Notifaction
Paramedics
submitsCase(recordsCasedetail)
accessDetailsCase(recordsCasedetail)
presentsCaseInformation(category:int, location:str, type)
getPrioritisedList (CrewID:str,state:boolean,location:content)
Alternative
{noavailable}
Alternative
{category=1 /2}
createAmbulanceNotify(nofitication:content)
sendNotifaction(AmbulanceNotify)
{else}
{else}
displayCanUseCrew(CrewID:str,location:content)
selectAvailable AmbulanceCrews(CrewID)
monitor(state)
sendDispatchCase (caseDetail)
presentsAcknowledgement(caseDetail)
logsCall-out(time, caseDetail, dispatcherID)

1) submitsCase(recordsCasedetail)
submits the case details (call, emergency location, category, decision tree)

pre-condition: The case already exists

post-condition: Get the details of the case, accessDetailsCase(recordsCasedetail) was triggered.

2) presentsCaseInformation(category:int, location:str, type)
the Dispatcher receives a case

pre-condition:the detail of the case was accessed by the dispatcher

post-condition:the system had already present the information of the case to the dispatcher, and then getPrioritisedList (CrewID:str,state:boolean,location:content) operation was triggered.

3) accessDetailsCase(recordsCasedetail)
they will access the details of the case

pre-condition:the case was gained by the dispatcher

post-condition:the dispatcher had already accessed the details of the case from the event access, and then presentsCaseInformation(category:int, location:str, type) operation was triggered.

4) getPrioritisedList (CrewID:str,state:boolean,location:content)
prioritised list of Ambulance Crews to the Dispatcher

Pre-condition:the system had already present the information of the case

Post-condition:the system had already gained the list from the ambulance crew management, and then createAmbulanceNotify(notification:content) operation was triggered

5) createAmbulanceNotify(nofitication:content)
The System presents the case information (category, location, type)

Pre-condition: the system had already gained the list from the ambulance crew management

Post-condition: ambulance crew management had already created the ambulance notify in the notification, and then sendNotifiy(ambulanceNotify) operation was triggered.

6) sendNotifaction(AmbulanceNotify)
Send the notification to paramedics

pre-condition:the ambulance notify had already created in the notification

post-condition:the paramedics had already obtained/received the relevant notification , and the system was display the available crew to the dispatcher, and then selectAvailable AmbulanceCrews(CrewID) operation was triggered

7) displayCanUseCrew(CrewID:str,location:content)
display the list of Ambulance Crews to the Dispatcher

pre-condition: Crew's information has been obtained post-condition: Dispatcher selects from the list, selectAvailable AmbulanceCrews(CrewID) was triggered

8) selectAvailable AmbulanceCrews(CrewID)
The System select the Ambulance Crew

pre-condition: the notification was got by the paramedics.

post-condition: the available crew was selected by the dispatcher, and then monitor(state) operation was triggered

9) sendDispatchCase (caseDetail)
The System send the DispatchCase to Ambulance Crew Management

pre-condition: the dispatcher had already monitored the status

post-condition:the system had already submit the case(including the case's detail) to the ambulance crew management,and then presentsAcknowledgement(caseDetail) operation was triggered

10) presentsAcknowledgement(caseDetail)
presents the acknowledgement of the case to the Dispatcher

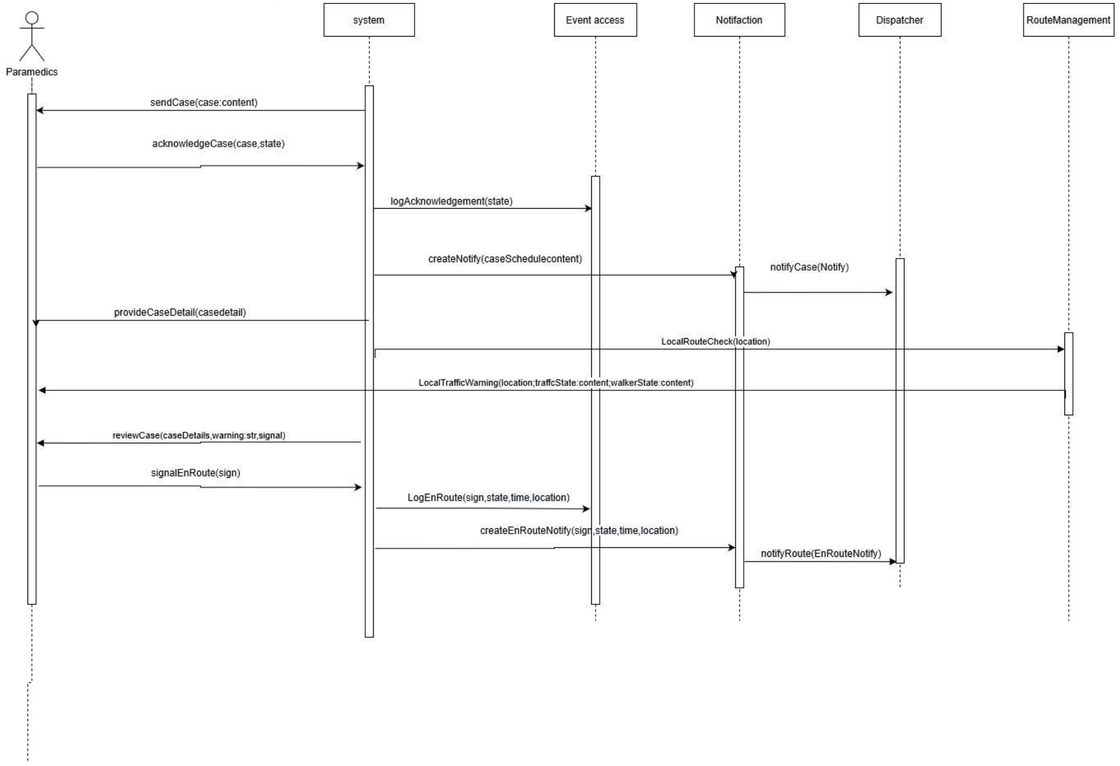
pre-condition: the system had already submit the case and ambulance crew management had already received the case

post-condition:the system had already displayed the acknowledgement ,and the system was recorded the call-out(including time, case's detail, and dispatcher ID number) in the event access, and then caseAcknowledge() operation was triggered.

11) logsCall-out(time, caseDetail, dispatcherID)
logs all detail(time, caseDetail, dispatcherID)

pre-condition: The case has been submitted to the dispatcher for confirmation

UC03: Acknowledge and Attend Case


Paramedics
system
Event access
Notifaction
Dispatcher
RouteManagement
sendCase(case:content)
acknowledgeCase(case,state)
logAcknowledgement(state)
createNotify(caseSchedulecontent)
notifyCase(Notify)
provideCaseDetail(casedetail)
LocalRouteCheck(location)
LocalTrafficWarning(location;traffcState:content;walkerState:content)
reviewCase(caseDetails,warning:str,signal)
signalEnRoute(sign)
LogEnRoute(sign,state,time,location)
createEnRouteNotify(sign,state,time,location)
notifyRoute(EnRouteNotify)

1) sendCase(case:content)
The system send the case to Ambulance Crew

Pre-conditions: The case already exists

Post-conditions: Send the case to the paramedics for confirmation, and the acknowledgecase (case, state) is triggered

2) acknowledgeCase(case,state)
Ambulance Crew acknowledge the case

Pre-conditions: the paramedics were received a case

Post-conditions: the system was recorded the acknowledged cases, and then createNotify(caseScheduleContent) operation was triggered.

3) logAcknowledgement(state)
The System logs the acknowledgement of the case

pre-condition: The system has confirmed the case

post-condition: The system will notify the dispatcher that createnotify (case schedule content) is triggered

4) createNotify(caseSchedulecontent)
Notification create the notify

pre-condition:the system was recorded the acknowledged case

post-condition:event access was created notify by the system , and then notifyCase(Notify) operation was triggered.

5) notifyCase(Notify)
notifies the Dispatcher of the case

pre-condition: the system was already created a notify in the notification

post-condition:the dispatchers had already knew the case, and then provideCaseDetail(caseDetail) was triggered.

6) provideCaseDetail(casedetail)
presents the Ambulance Crew with the details of the case

Pre-condition: the dispatcher has been notified (the case)

Post-condition: the paramedics were gained the detail of the case ,and then localRouteCheck(location:string) was triggered.

7) LocalRouteCheck(location)
Check the local traffic conditions

pre-condition:the paramedics were gained the case's detail

post-condition:the system had already obtained the local routes, and then localTraiffcWarning(location,trafficStatus:content,walkerState:content)

operation was triggered.

8) LocalTrafficWarning(location;traffcState:content;walkerState:content)
presents the warnings of local traffic conditions

Pre-condition: the route was checked by the route management

Post-condition:the paradmedics was get the local traffic warning, and then signalEnRoute(sign)operation was triggered.

9) reviewCase(caseDetails,warning:str,signal)
The Ambulance Crew reviews the case details

Pre-condition: Traffic warning has been received

Post-condition: The first responder sends a signal halfway and the signalenroute (sign) is triggered

10) signalEnRoute(sign)
signals that they are en route to the case

pre-condition:the traffic warning was gained by the paramedics ,the paramedics were reviewed the case.

post-condition:the system had already recorded the paramedics were En Route, and then createEnRouteNotify(sign,state,location,time) was triggered

11) LogEnRoute(sign,state,time,location)
logs that the Ambulance Crew is en route

Pre-condition: Medical staff are already on their way

Post-condition: After all details are recorded, createenroutenotify (sign, state, time, location) is triggered

12) createEnRouteNotify(sign,state,time,location)
Create the notify about they are en route to the case

pre-condition: the EnRoute has been logged

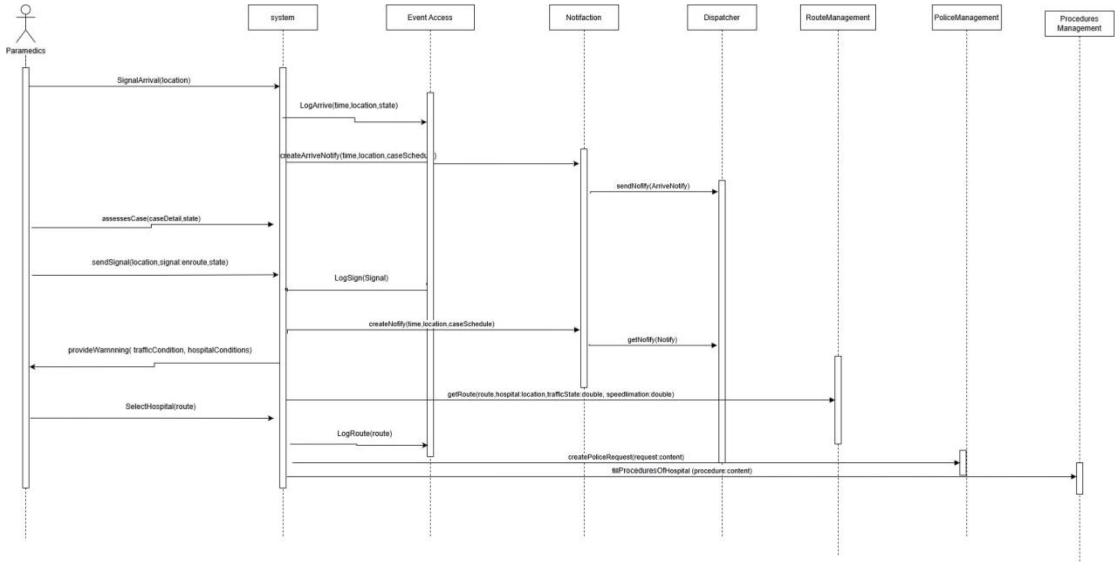
post-condition:the En Route notification was created in the notification, and then notifyRoute(EnRouteNotify) was triggered

13) notifyRoute(EnRouteNotify)
notifies the Dispatcher

pre-condition:En Route notification was created in the notification

post-condition:the dispatcher had already gained the notification about route, and then signalArrival(location) was triggered.

UC04: Assess Case


Paramedics
system
Event Access
Notifaction
Dispatcher
RouteManagement
PoliceManagement
Procedures Management
SignalArrival(location)
LogArrive(time,location,state)
createArriveNotify(time,location,caseSchedu)
sendNofify(ArriveNotify)
assessesCase(caseDetail,state)
sendSignal(location,signal:enroute,state)
LogSign(Signal)
createNofify(time,location,caseSchedule)
getNofify(Notify)
provideWarnnning(trafficCondition, hospitalConditions)
getRoute(route,hospital:location,trafficState:double, speedlimation:double)
SelectHospital(route)
LogRoute(route)
createPoliceRequest(request:content)
fillProceduresOfHospital (procedure:content)

1) SignalArrival(location)
Signal the Ambulance Crew arrival at the emergency location of the case

pre-condition: the dispatcher had already gained the notification about route

post-condiition:the system had already record/log arrival information(including time,location and state), and then createArrivalNotify(time,location,caseSchedule) operation was triggered.

2) LogArrive(time,location,state)
The System logs the arrival of the Ambulance Crew

pre-condition: The arrival signal has been received

post-condiition: When an arrival reminder is created, createarrevenotify (time, location, caseschedule) is triggered

3) createArriveNotify(time,location,caseSchedule)
create the notify about the Ambulance Crew arrival

pre-condition:arrival information was recorded /logged by the system

post-condition:Notification had already created arrival notify(including time ,location.ect), and then sendNotify(ArrivalNotify) operation was triggered.

4) sendNofify(ArriveNotify)
Send notifies the Dispatcher

pre-condition: the notification had already created arrival notify post-condition:the dispatcher had already received the notify(arrival notify), and then accessesCase(caseDetail,state) operation was triggered.

5) assessesCase(caseDetail,state)
Ambulance Crew assesses the case

pre-condition:notify(arrival notify) was received by the dispatcher

post-condition:the case had already accessed by the dispatcher,and then sendSignal(location,signal,enroute,state) operation was triggered.

6) sendSignal(location,signal:enroute,state)
signals that they are en route to the hospital

pre-condition: the case had already accessed by the dispatcher

post-condition:the system was recorded/logged the signal, and then createNotify(time,location,caseSchedule) operation was triggered.

7) LogSign(Signal)
The system logs that the Ambulance Crew is headed to a hospital

pre-condition: System received signal

post-condition: After receiving the signal, you need to send a notify, and then createNotify(time,location,caseSchedule) operation was triggered.

8) createNotify(time,location,caseSchedule)
Create the notify

pre-condition:the signal had already recorded by the system

post-condition:the notification had already created the notify(time,location,caseSchedule),and then getNotify(Notify) was triggered.

9) getNofify(Notify)
notifies the Dispatcher

pre-condition:the notify was already created in the notification

post-condition:the dispatcher was gained the notify , and then provideWarning(trafficCondition,hospitalConditions) operation was triggered

10) provideWarning(trafficCondition, hospitalConditions)

Presents any warnings about traffic, hospital conditions to the Ambulance Crew

Pre-condition:the notify was obtained by the dispatchers

Post-condition:the paramedics had already received the warning(including traffic condition and hospital condition) , and then getRoute(route,hospital:location,trafficState:double,sppedLimitation:double) operation was triggered

11) getRoute(route,hospital:location,trafficState:double, speedlimation:double)
Get the detail about traffic, hospital conditions

pre-condition:the warning was received by the paramedics

post-condition:the system had already gained the route, and the route was logged into the event access ,and then selectHospital(route) operation was triggered

12) SelectHospital(route)
Ambulance Crew reviews the information, decides on the hospital

pre-condition: the route was logged into the event access

post-conditon:the paramedics had already selected the hospital to send the patient, and then createPolicyRequest(request:content) operation was triggered.

13) LogRoute(route)
Logs the route

pre-condition: The destination hospital has been selected

post-conditon: After the route log is completed, the police need to be notified and the createpolicerequest (request: content) is triggered

14) createPoliceRequest(request:content)
The System presents any policies

pre-condition:the hospital was selected by the paramedics

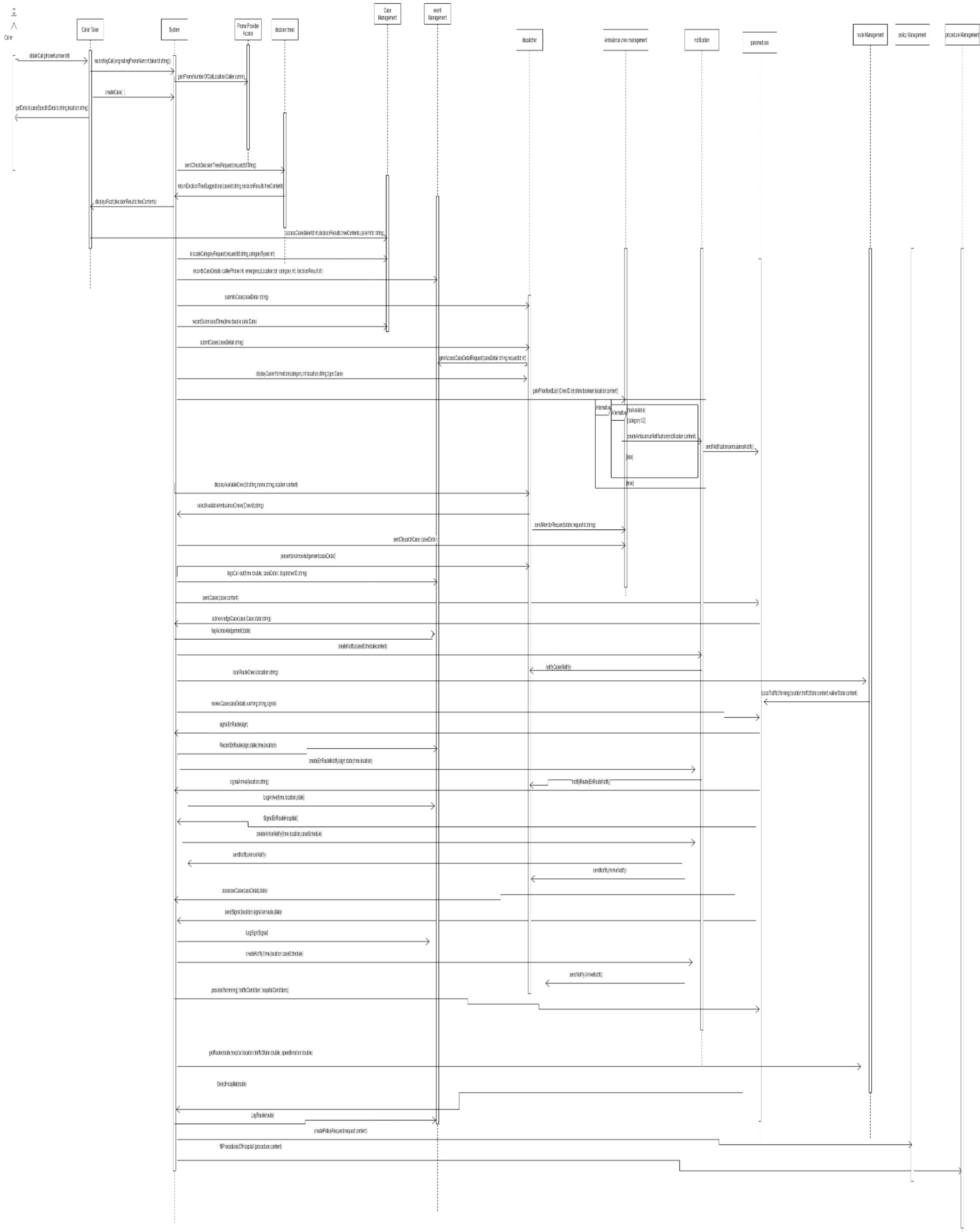
post-condition:the system had already displayed the policies, and then fillProcedureOfHospital(procedure:content) operation was triggered.

15) fillProceduresOfHospital (procedure:content)
procedures to be reviewed while in transit

pre-condition:the system had already displayed the policies.

post-condition: the procedure was filled into the procedure management from the system.

6. Component Interfaces


Caller
Caller Taker
System
Phone Provider Access
decision trees
Case Management
event Management
dispatcher
Ambulance crew management
notification
paramedics
route Management
policy Management
procedure Management
obtainCall(phoneNumber:int)
recordlogCall(originatingPhoneNum:int,takerId:string()
gainPhoneNumberOfCallLocation(Caller.comm)
createCase( )
getDetails(caseSpecificDetails:string,location:string()
sendCheckDecisionTreesRequest(requestId:string(
returnDecisionTreeSuggestions(caseId:string,decisionResults:treeContents
displaysRoot(decisionResults:treeContents)
accessCase(takerId:int,decisionResults:treeContents,callerInfor:string()
allocateCategoryRequest(requestId:string,categoryTypes:int()
recordsCaseDetails (callerPhone:int, emergencyLocation:str, category:int, decisionResult:str)
submitsCase(caseDetail:string)
recordSubmissedTime(time:double,date:Date)
submitCases(caseDetail:string(
sendAccessCaseDetailRequest(caseDetail:string,requestId:int(
displayCaseInformation(category:int,location:string,type:Case)
gainPrioritisedList (CrewID:str,state:boolean,location:content()
Alternative
Alternative
[noAvailable]
[category1/2]
createAmbulanceNotification(notification:content)
sendNotification(ambulanceNotify)
[else]
[else]
displayAvailableCrew(id:string,name:string,location:content)
selectAvailableAmbulanceCrews(CrewId:string)
sendMonitorRequest(state:requestId:string)
sendDispatchCase (caseDetail)
presentsAcknowledgement(caseDetail()
logsCall-out(time:double, caseDetail, dispatcherID:string)
sendCases(case:content)
acknowledgeCase(case:Case,state:string)
logAcknowledgement(state)
createNotify(caseSchedule:content)
notifyCase(Notify)
localRouteCheck(location:string)
LocalTrafficWarning(location,trafficState:content,walkerState:content()
reviewCase(caseDetails,warning:string,signal)
signalEnRoute(sign(
RecordEnRoute(sign,state,time,location)
createEnRouteNotify(sign,state,time,location)
signalArrival(location:string(
notifyRoute(EnRouteNotify()
LogArrive(time,location,state)
SignalEnRouteHospital()
createArriveNotify(time,location,caseSchedule)
sendNotify(ArriveNotify)
sendNotify(ArriveNotify)
assessesCase(caseDetail,state)
sendSignal(location,signal:enroute,state)
LogSign(Signal(
createNotify(time,location,caseSchedule)
sendNotify(ArriveNotify)
provideWarmning(trafficCondition, hospitalConditions)
getRoute(route,hospital:location,trafficState:double, speedlimation:double)
SelectHospital(route)
LogRoute(route)
createPoliceRequest(request:content)
fillProceduresOfHospital (procedure:content)

Operation contract:

1) operation:getCall(phoneNumber:int)
getCall(phoneNumber:int) : to receive the call from caller

pre-condition:caller has called to the caller taker

post-condition: caller taker had already obtained the call, and the system had already record the call((TakerID,originatingPhone), and then getCallerLocation(call:comm) operation was triggered.

2) operation: getCallerLocation(call:comm)
getCallerLocation(call:comm) : to get the location from the phone provider

Pre-condition:the call was received by the caller taker.

Post-condition:the system was obtained the accurate location from the telephone provider, and then getDetail(location:str,caseDetail:str) operation was triggered

3)operation: getDetail(location:str,caseDetail:str)

getDetail(location:str,caseDetail:str):get the specific detail from the caller taker

pre-condition:the system had already gained the location from the phone provider

post-condition: the caller had already got the detail from the caller taker, and then getDecisionTreeSuggest(caseID:str, decisionResult:content) operation was triggered

4)operation: getDecisionTreeSuggest(caseID:str, decisionResult:content)

getDecisionTreeSuggest(caseID:str, decisionResult:content): to get the relevant decision trees suggestions from decision tree.

Pre-condition:the caller was received the detail

Post-condition:the suggestions(decision tree) had already gained by the system, and then presentsRoot(decisonResult:content) operation was triggered.

5)operation: presentsRoot(decisonResult:content)

presentsRoot(decisonResult:content):

pre-condition:the system was got the suggestions from the decision tree

post-condition:the system had already displayed the root to the caller taker, and then assessesCase(takerID:str,callerInformation:str,decisonResult:content) operation was triggered.

6)operation: assessesCase(takerID:str,callerInformation:str,decisonResult:content)

assessesCase(takerID:str,callerInformation:str,decisonResult:content): is to access the case from the case management pre-condition: the system had already displayed the root to the caller taker

post-condition: the case had already accessed by the case management, and then allocatesCategory(categoryType:int) operation was triggered

7)operation: allocatesCategory(categoryType:int)

allocatesCategory(categoryType:int): get the category levels(category1, category2, ect) from the case management

pre-condition: the case had already accessed by the case management

post-condition:the system had gained the category type from the case management, the system was recorded the case details in the event access,and then submitsCase(recordsCasedetail) operation was triggered.

8) operation :submitsCase(recordsCasedetail)

submitsCase(recordsCasedetail) : to submit the case(including the case's details) to the dispatcher

pre-condition: the details of the case was recorded by the system

post-condition:the dispatcher had already received the case, and the system was recorded the submission time to the case management, and then accessDetailsCase(recordsCasedetail) operation was triggered.

9) operation: accessDetailsCase(recordsCasedetail)

accessDetailsCase(recordsCasedetail): to access the details from the event access

pre-condition:the case was gained by the dispatcher

post-condition:the dispatcher had already accessed the details of the case from the event access, and then presentsCaseInformation(category:int, location:str, type) operation was triggered.

10)operation: presentsCaseInformation(category:int, location:str, type)

presentsCaseInformation(category:int, location:str, type): is to show the information of the case to the dispatcher

pre-condition:the detail of the case was accessed by the dispatcher

post-condition:the system had already present the information of the case to the dispatcher, and then getPrioritisedList (CrewID:str,state:boolean,location:content) operation was triggered.

11)operation: getPrioritisedList (CrewID:str,state:boolean,location:content)

Pre-condition:the system had already present the information of the case

Post-condition:the system had already gained the list from the ambulance crew management, and then createAmbulanceNotify(notification:content) operation was triggered

12)operation: createAmbulanceNotify(notification:content)

Pre-condition: the system had already gained the list from the ambulance crew management

Post-condition: ambulance crew management had already created the ambulance notify in the notification, and then sendNotifiy(ambulanceNotify) operation was triggered.

13)operation:sendNotify(ambulanceNotify)

sendNotify(ambulanceNotify): to send the notification to the paramedics

pre-condition:the ambulance notify had already created in the notification

post-condition:the paramedics had already obtained/received the relevant notification , and the system was display the available crew to the dispatcher, and then selectAvailable AmbulanceCrews(CrewID) operation was triggered

14) operation: selectAvailable AmbulanceCrews(CrewID)

selectAvailable AmbulanceCrews(CrewID): is to choose the available crew from the ambulance crew management

pre-condition: the notification was got by the paramedics.

post-condition: the available crew was selected by the dispatcher, and then monitor(state) operation was triggered

15) operation: monitor(state)

Pre-condition: the available crew was selected by the dispatcher

Post-condition: the dispatcher had already monitored the status from the ambulance crew management, and then sendDispatchCase (caseDetail) operation was triggered.

16) operation: sendDispatchCase (caseDetail)

sendDispatchCase (caseDetail): to send the case to the dispatcher

pre-condition: the dispatcher had already monitored the status

post-condition:the system had already submit the case(including the case's detail) to the ambulance crew management,and then presentsAcknowledgement(caseDetail) operation was triggered

17) operation: presentsAcknowledgement(caseDetail)

presentsAcknowledgement(caseDetail) : to display the acknowledgement to the dispatchers pre-condition: the system had already submit the case and ambulance crew management had already received the case

post-condition:the system had already displayed the acknowledgement ,and the system was recorded the call-out(including time, case's detail, and dispatcher ID number) in the event access, and then caseAcknowledge() operation was triggered.

18) Operation:caseAcknowledged()

caseAcknowledge() is to acknowledge the case by the paramedics

Pre-conditions: the paramedics were received a case

Post-conditions: the system was recorded the acknowledged cases, and then createNotify(caseScheduleContent) operation was triggered.

19)Operation: createNotify(caseScheduleContent)

createNotify(caseScheduleContent) is to create a notify in the notification

pre-condition:the system was recorded the acknowledged case

post-condition:event access was created notify by the system , and then notifyCase(Notify) operation was triggered.

20)Operation:notifyCase(Notify)

notifyCase(Notify) is to notify the cases to the dispatcher.

pre-condition: the system was already created a notify in the notification

post-condition:the dispatchers had already knew the case, and then provideCaseDetail(caseDetail) was triggered.

21)Operation:provideCaseDetail(caseDetail)

provideCaseDetail(caseDetail) : to offer/provide the detail of case to the paramedics.

Pre-condition: the dispatcher has been notified (the case)

Post-condition: the paramedics were gained the detail of the case ,and then localRouteCheck(location:string) was triggered.

22)operation:localRouteCheck(location:string)

localRouteCheck(location:string) : to check the traffic condition from the route management pre-condition:the paramedics were gained the case's detail

23)post-condition:the system had already obtained the local routes, and then localTraiffcWarning(location,trafficStatus:content,walkerState:content)

operation was triggered.

24)Operation:localTraiffcWarning(location,trafficStatus:content,walkerState:content)

localTraiffcWarning(location,trafficStatus:content,walkerState:content): to get the traffic warnings from the route management.

Pre-condition: the route was checked by the route management

Post-condition:the paradmedics was get the local traffic warning, and then signalEnRoute(sign)operation was triggered.

25)operation:signalEnRoute(sign) :

signalEnRoute(sign) is to send signal (En route) to the system

pre-condition:the traffic warning was gained by the paramedics ,the paramedics were reviewed the case.

post-condition:the system had already recorded the paramedics were En Route, and then createEnRouteNotify(sign,state,location,time) was triggered

26)operation: createEnRouteNotify(sign,state,location,time)

createEnRouteNotify(sign,state,location,time) is to create En Route notify to notification

pre-condition:

post-condition:the En Route notification was created in the notification, and then notifyRoute(EnRouteNotify) was triggered

27)operation: notifyRoute(EnRouteNotify)

pre-condition:En Route notification was created in the notification

post-condition:the dispatcher had already gained the notification about route, and then signalArrival(location) was triggered.

28)operation: signalArrival(location)

signalArrival(location) is to

pre-condition: the dispatcher had already gained the notification about route

post-condiition:the system had already record/log arrival information(including time,location and state), and then createArrivalNotify(time,location,caseSchedule) operation was triggered.

29)operation:createArrivalNotify(time,location,caseSchedule)

createArrivalNotify(time,location,caseSchedule) is to send arrival notify to the notification

pre-condition:arrival information was recorded /logged by the system

post-condition:Notification had already created arrival notify(including time ,location.ect), and then sendNotify(ArrivalNotify) operation was triggered.

30)operation: sendNotify(ArrivalNotify)

sendNotify(ArrivalNotify) is to send the notify from the notification to the dispatchers

pre-condition: the notification had already created arrival notify

post-condition:the dispatcher had already received the notify(arrival notify), and then accessesCase(caseDetail,state) operation was triggered.

31)operation:accessesCase(caseDetail,state)

accessesCase(caseDetail,state) : to access the case by the paramedics

pre-condition:notify(arrival notify) was received by the dispatcher post-condition:the case had already accessed by the dispatcher,and then sendSignal(location,signal,enroute,state) operation was triggered.

32)operation: sendSignal(location,signal,enroute,state)

sendSignal(location,signal,enroute,state) :to send signal from the paramedics to the system

pre-condition: the case had already accessed by the dispatcher

post-condition:the system was recorded/logged the signal, and then createNotify(time,location,caseSchedule) operation was triggered.

33)Operation: createNotify(time,location,caseSchedule)

createNotify(time,location,caseSchedule): create the notify from the system to the notification

pre-condition:the signal had already recorded by the system

post-condition:the notification had already created the notify(time,location,caseSchedule),and then getNotify(Notify) was triggered.

34)Operation:getNotify(Notify)

getNotify(Notify):the dispatcher receive the notify from notification

pre-condition:the notify was already created in the notification

post-condition:the dispatcher was gained the notify , and then provideWarning(trafficCondition,hospitalConditions) operation was triggered

35)Operation:provideWarning(trafficCondition,hospitalConditions)

provideWarning(trafficCondition,hospitalConditions): to send warnings(like which sections of road has traffic jam or traffic accidents, which hospital is available, ect) to the dispatchers.

Pre-condition:the notify was obtained by the dispatchers

Post-condition:the paramedics had already received the warning(including traffic condition and hospital condition) , and then getRoute(route,hospital:location,trafficState:double,sppedLimitation:double) operation was triggered

36) operation: getRoute(route, hospital:location,trafficState:double,sppedLimitation:double)

getRoute(route, hospital:location,trafficState:double,sppedLimitation:double) :get relevant route from the route management

pre-condition:the warning was received by the paramedics

post-condition:the system had already gained the route, and the route was logged into the event access ,and then selectHospital(route) operation was triggered

37) operation: selectHospital(route)

selectHospital(route): the paramedics select the hospital to send the patient
pre-condition: the route was logged into the event access

post-conditon:the paramedics had already selected the hospital to send the patient, and then createPolicyRequest(request:content) operation was triggered.

38) Operation:createPolicyRequest(request:content)
createPolicyRequest(request:content):get the relevant policy from the policy management
pre-condition:the hospital was selected by the paramedics
post-condition:the system had already displayed the policies, and then fillProcedureOfHospital(procedure:content) operation was triggered.

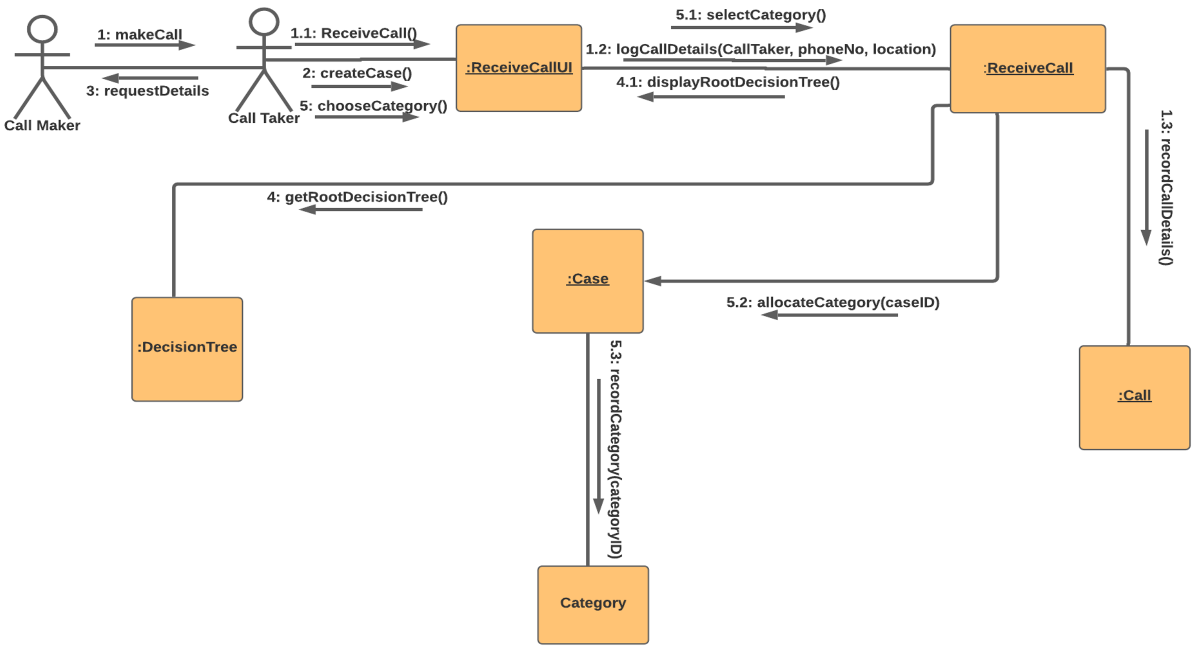
39) Operation: fillProcedureOfHospital(procedure:content)
pre-condition:the system had already displayed the policies.

post-condition: the procedure was filled into the procedure management from the system.

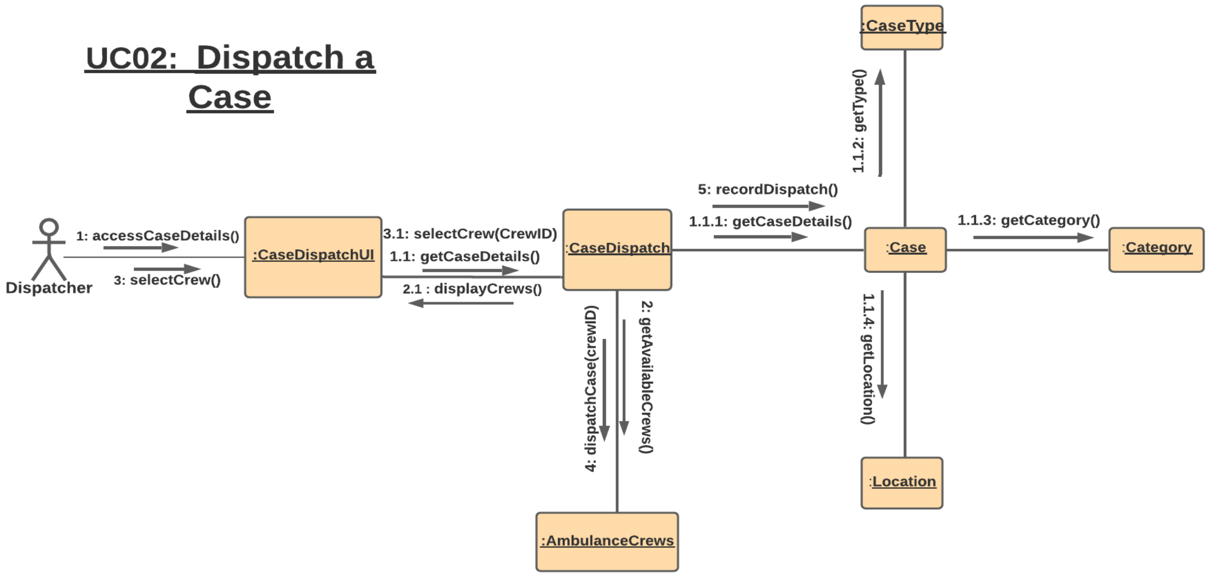
7. Implementation Design

Communication Diagrams

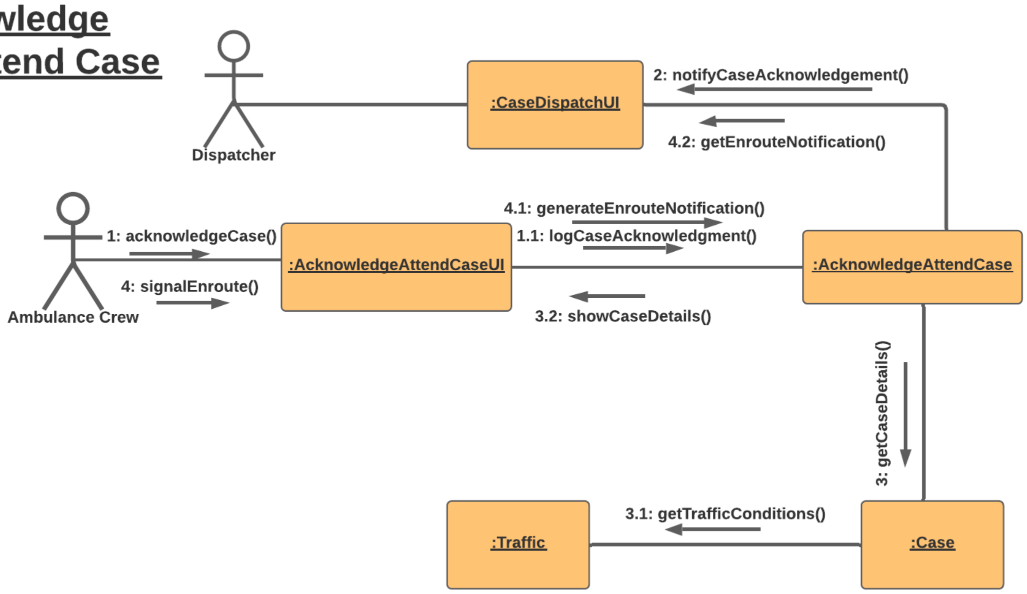
UC01: Receive Emergency Call



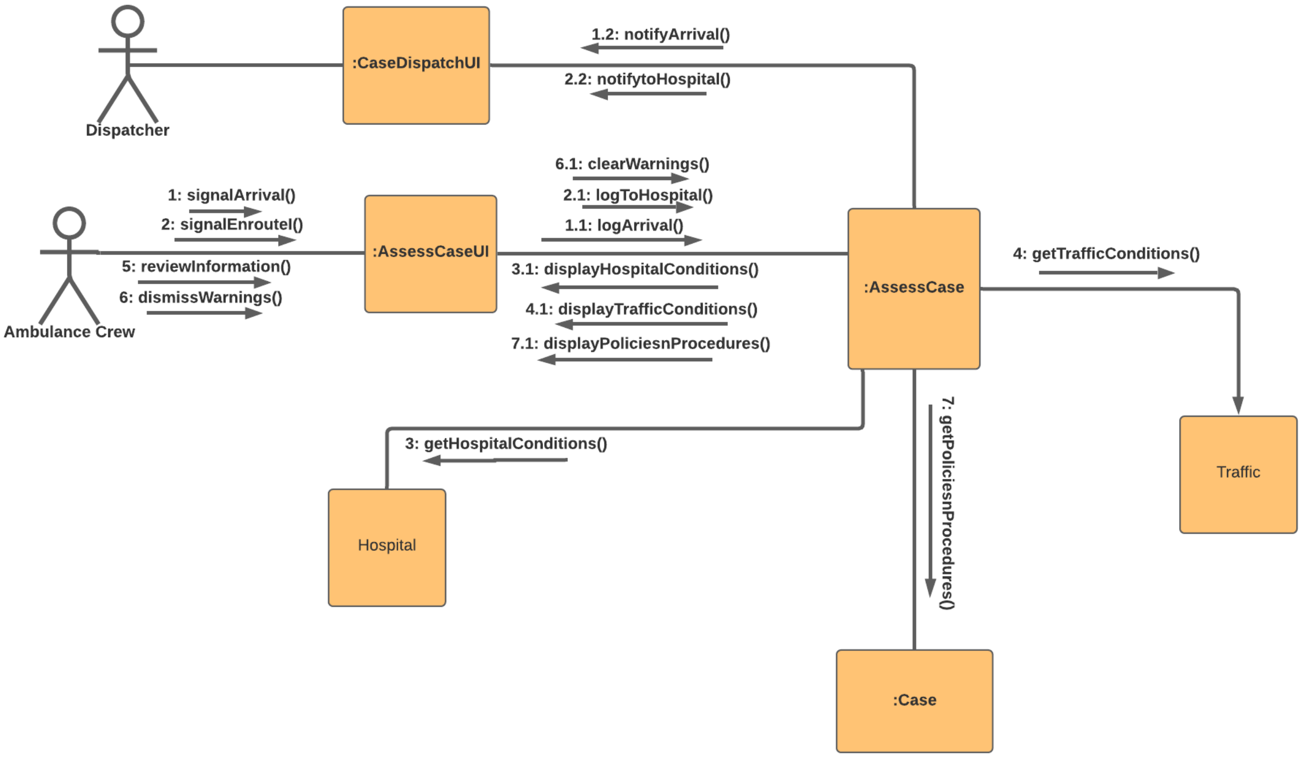
UC02: Dispatch a Case



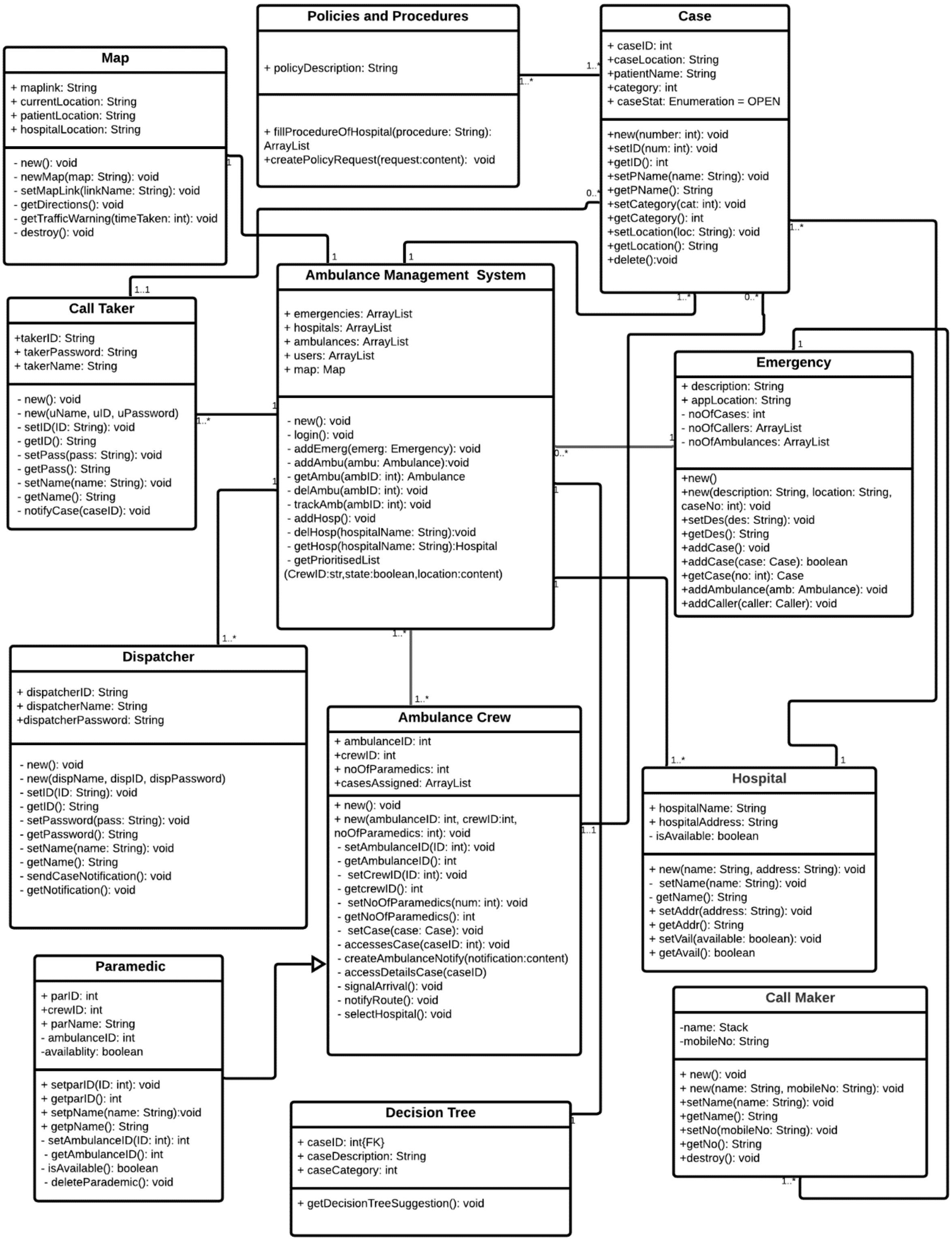
UC03: Acknowledge and Attend Case



UC04: Assess Case

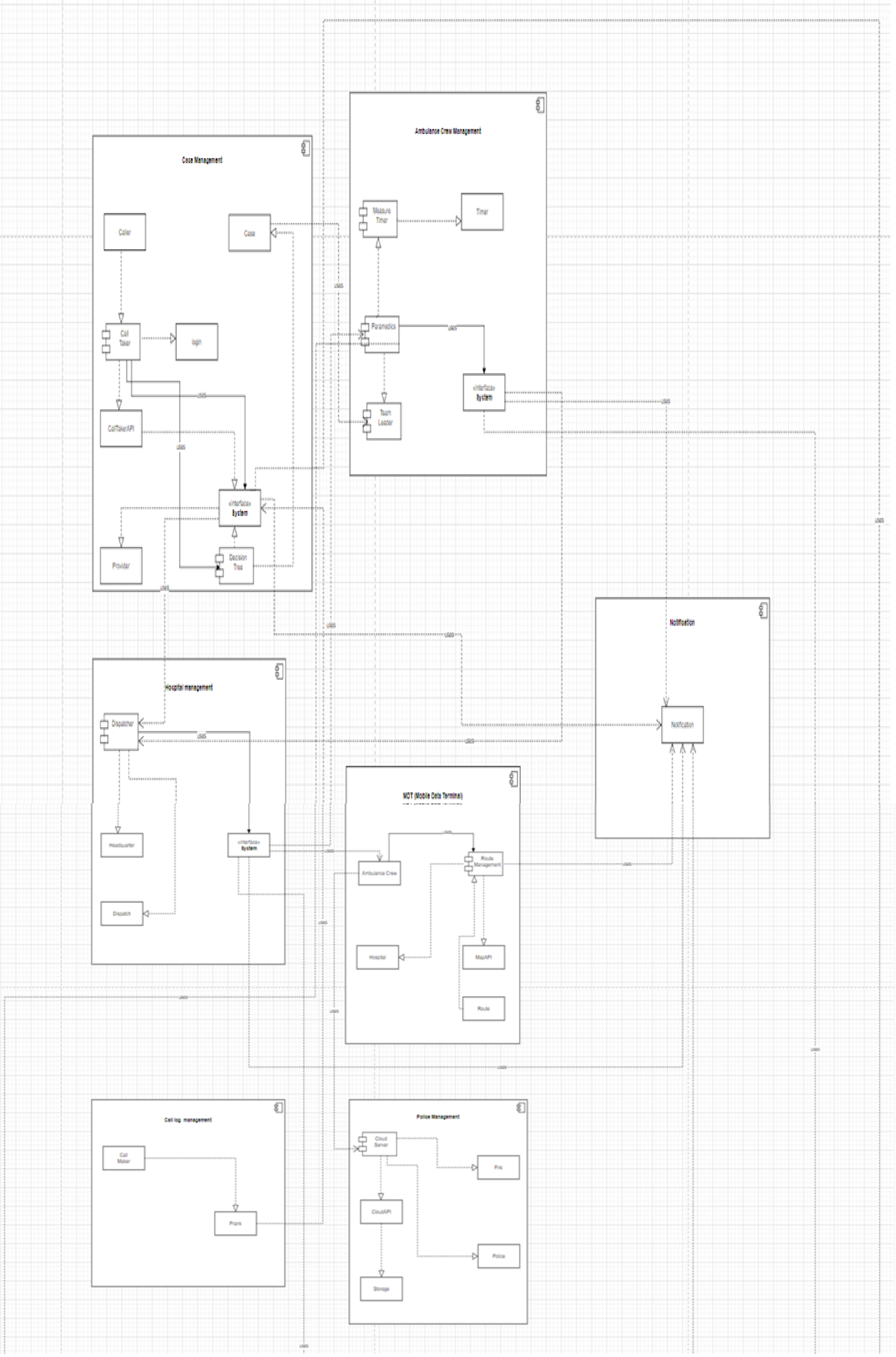


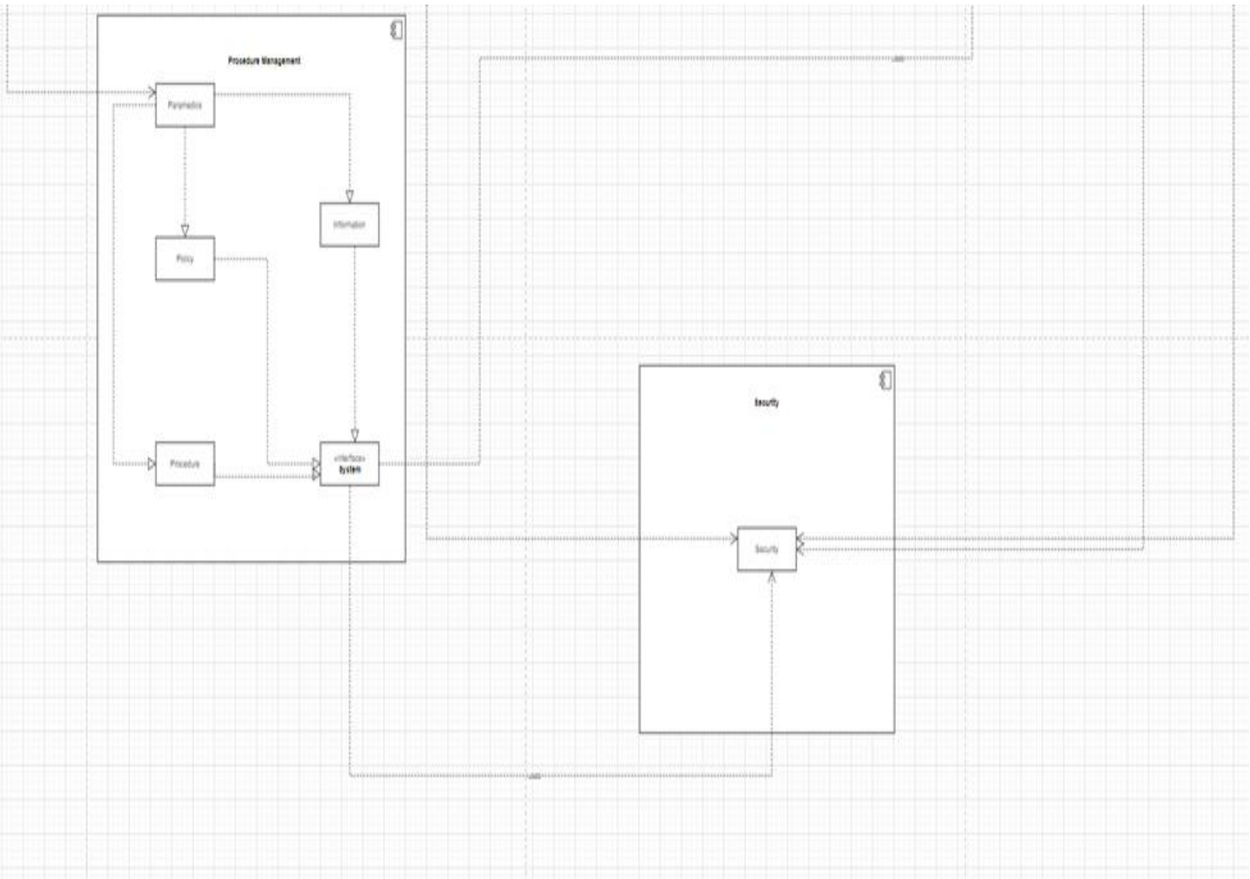
UML Class Diagram



Policies and Procedures
+ policyDescription: String
+ fillProcedureOfHospital(procedure: String): ArrayList
+createPolicyRequest(request:content): void
Case
+ caseID: int
+caseLocation: String
+patientName: String
+category: int
+ caseStat: Enumeration = OPEN
+new(number: int): void
+setID(num: int): void
+getID(): int
+setPName(name: String): void
+getPName(): String
+setCategory(cat: int): void
+getCategory(): int
+setLocation(loc: String): void
+getLocation(): String
+delete():void
Map
+ maplink: String
+ currentLocation: String
+ patientLocation: String
+ hospitalLocation: String
- new(): void
- newMap(map: String): void
- setMapLink(linkName: String): void
- getDirections(): void
- getTrafficWarning(timeTaken: int): void
- destroy(): void
Call Taker
+takerID: String
+ takerPassword: String
+ takerName: String
- new(): void
- new(uName, uID, uPassword)
- setID(ID: String): void
- getID(): String
- setPass(pass: String): void
- getPass(): String
- setName(name: String): void
- getName(): String
- notifyCase(caseID): void
Ambulance Management System
+ emergencies: ArrayList
+ hospitals: ArrayList
+ ambulances: ArrayList
+ users: ArrayList
+ map: Map
- new(): void
- login(): void
- addEmerg(emerg: Emergency): void
- addAmbu(ambu: Ambulance):void
- getAmbu(ambID: int): Ambulance
- delAmbu(ambID: int): void
- trackAmb(ambID: int): void
- addHosp(): void
- delHosp(hospitalName: String):void
- getHosp(hospitalName: String):Hospital
- getPrioritisedList (CrewID:str,state:boolean,location:content)
Emergency
+ description: String
+ appLocation: String
- noOfCases: int
- noOfCallers: ArrayList
- noOfAmbulances: ArrayList
+new()
+new(description: String, location: String, caseNo: int): void
+setDes(des: String): void
+getDes(): String
+addCase(): void
+addCase(case: Case): boolean
+getCase(no: int): Case
+addAmbulance(amb: Ambulance): void
+addCaller(caller: Caller): void
Dispatcher
+ dispatcherID: String
+ dispatcherName: String
+dispatcherPassword: String
- new(): void
- new(dispName, dispID, dispPassword)
- setID(ID: String): void
- getID(): String
- setPassword(pass: String): void
- getPassword(): String
- setName(name: String): void
- getName(): String
- sendCaseNotification(): void
- getNotification(): void
Ambulance Crew
+ ambulanceID: int
+crewID: int
+ noOfParamedics: int
+casesAssigned: ArrayList
+ new(): void
+ new(ambulanceID: int, crewID:int, noOfParamedics: int): void
- setAmbulanceID(ID: int): void
- getAmbulanceID(): int
- setCrewID(ID: int): void
- getcrewID(): int
- setNoOfParamedics(num: int): void
- getNoOfParamedics(): int
- setCase(case: Case): void
- accessesCase(caseID: int): void
- createAmbulanceNotify(notification:content)
- accessDetailsCase(caseID)
- signalArrival(): void
- notifyRoute(): void
- selectHospital(): void
Hospital
+ hospitalName: String
+ hospitalAddress: String
- isAvailable: boolean
+ new(name: String, address: String): void
- setName(name: String): void
- getName(): String
+ setAddr(address: String): void
+ getAddr(): String
+ setVail(available: boolean): void
+ getAvail(): boolean
Paramedic
+ parID: int
+crewID: int
+ parName: String
- ambulanceID: int
-availablity: boolean
+ setparID(ID: int): void
+ getparID(): int
+ setpName(name: String):void
+ getpName(): String
- setAmbulanceID(ID: int): int
- getAmbulanceID(): int
- isAvailable(): boolean
- deleteParademic(): void
Call Maker
-name: Stack
-mobileNo: String
+ new(): void
+ new(name: String, mobileNo: String): void
+setName(name: String): void
+getName(): String
+setNo(mobileNo: String): void
+getNo(): String
+destroy(): void
Decision Tree
+ caseID: int{FK}
+ caseDescription: String
+ caseCategory: int
+ getDecisionTreeSuggestion(): void
1..*
1..*
1
0..*
1..*
1
1
1..1
1..*
0..*
1
1..*
1
1
0..*
1
1
1
1..*
1..*
1..1
1..*
1
1
1..*

8. Designing Boundaries


Case Management
Caller
Case
Call Taker
login
CallTakerAPI
«interface» System
Provider
Decision Tree
Ambulance Crew Management
Measure Timer
Timer
Paramedics
«interface» System
Team Leader
Notification
Notification
Hospital management
Dispatcher
Headquarter
«interface» System
Dispatch
MDT (Mobile Data Terminal)
Ambulance Crew
Route Management
Hospital
MapAPI
Route
Call log management
Call Maker
Prank
Police Management
Cloud Server
Fire
CloudAPI
Police
Storage
uses



This diagram it will going to separate in to nice class, case management, hospital management, ambulance crew management MDT (Mobile Data Terminal),call log management,police management,

Procedure Management Notification, security.

This system is start from the call taker and it separate in to two chose login and calTakerAPI for him to choose for running the system user to run, the call taker and provider will be going to make down all the detail and the hospital management,

If they are call are prank or criminal, call log management will going to decide how to solve it, if it is about criminal, the system will be going to send police.

Police management will going save it in the cloud server of police management and storage it, if it fires the system will told it to fire fighter, if it is criminal it will pass to police will going to solve it.

then the hospital management dispatcher will be going to get the case to the headquarter to classification the case. Hospital management will be going to put all the information just like name of patient, patient condition.

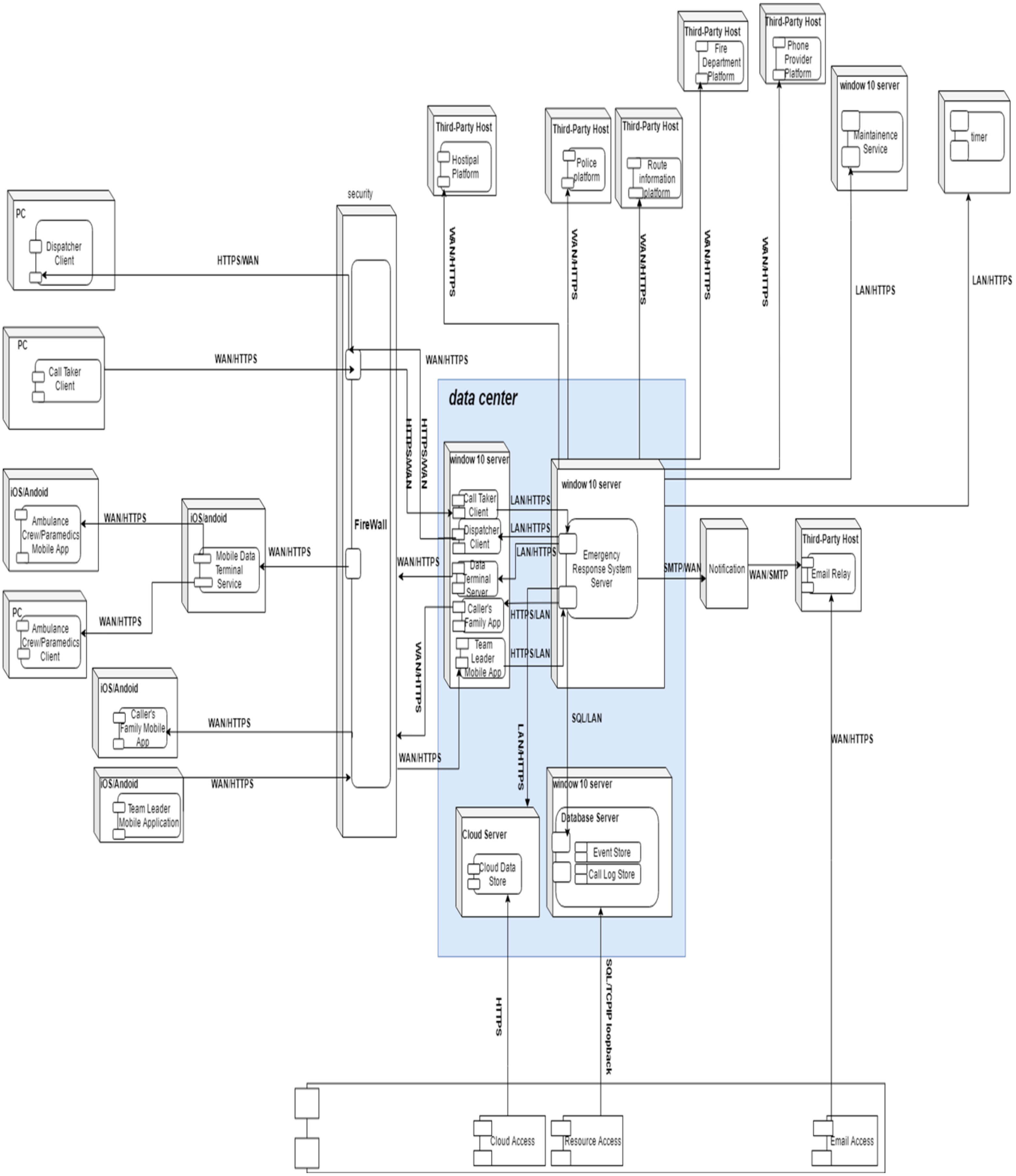
Mobile data terminal will be going to show all the paramedics name of different case, to let ambulance know where they should go to take after that ambulance crew will going to know the name of the case. And let them have enough detail of the patient.

In the notification past, the system will be going to collect all the detail of caller, ambulance crew, paramedics, to let them have a communicate in this area.

Procedure management will be going to get the information in the notice, they may get some information in there if they get the permit because of the security, if they don't have permit the policy will not going to give the information to them because of the security.

After classification the case will pass it to ambulance crew management part, procedures management are going to arrange, it will show all who is free, who are the near and who will going to get the case.

9. Deployment Diagram



There are few things to explain in this diagram:

1) Caller’s family mobile app: the system sends the patient’s current status (like coma) to this app. so caller’s family can view the patient’s status in real time on this app

2) Hospital platform: when the patient is on the way from the ambulance to the hospital, the crew can click the hospital platform button on the ambulance crew/paramedics mobile App to upload relevant cases, and the hospital can know the patient’s situation in advance.

It also can send back the treatment results of patients to the system server.

3) Route information platform: it currently displays the traffic conditions (whether there is a traffic jam or road closure or a traffic accident) on the main road. Ambulance crew/ paramedics can click this platform on the ambulance crew mobile Client to avoid these sections of the road, which enable patients to be taken/sent to hospital more quickly.

And Paramedics also get which path is the nearest

4) Fire department platform: after caller taker confirmed that the case is a fire case, he/she will immediately click the fire department platform button on the call taker Client and then upload the case into it, and finally the fire department will be able to deal with that case more rapidly.

Third-party hosts(like Fire department platform, Route information platform and hospital platform) are all interfaces in the system, so the system needs to transmit data with them through the WAN.