M&N19/4/COMSC/HP3/ENG/TZ0/XX/CS

Diploma Programme
Programme du diplôme
Programa del Diploma

Computer science
Higher level
Paper 3 – case study: a new computer aided dispatch system for Bangbai

For use in May and November 2019

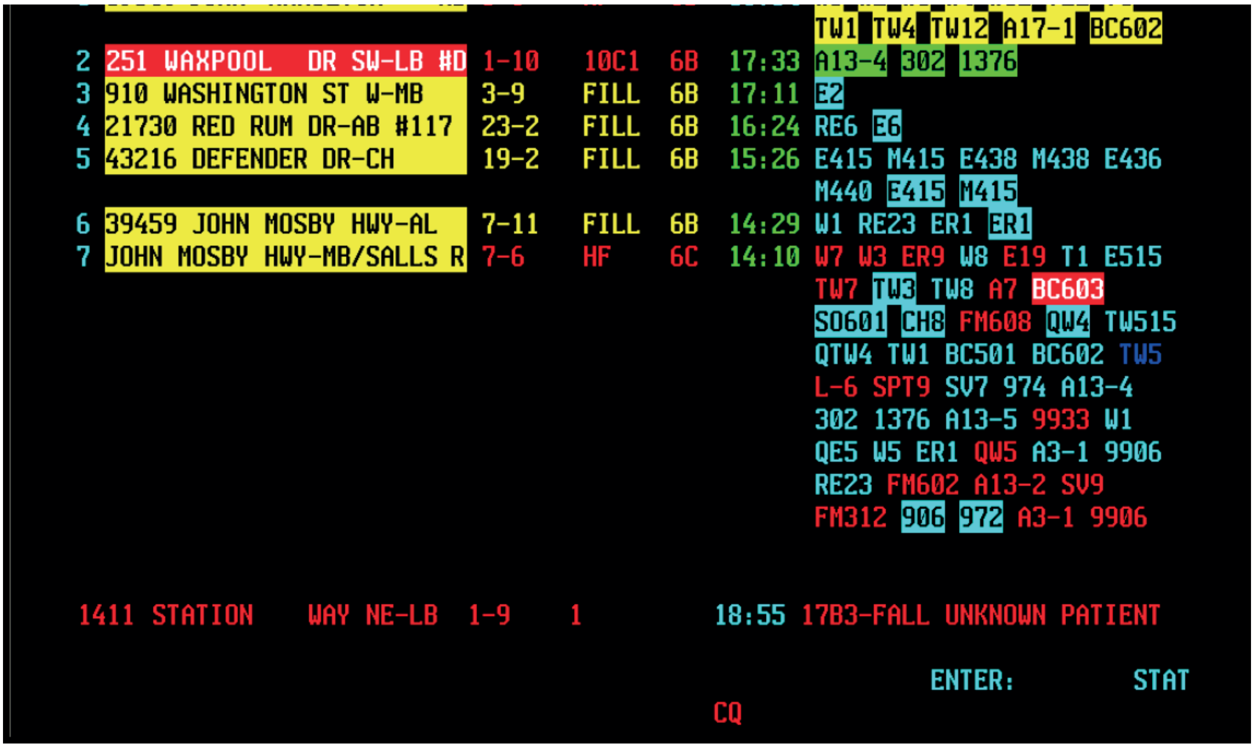
Instructions to candidates

- Case study booklet required for higher level paper 3.

Introduction

Since the introduction of the “999” *emergency number* in 1937, many countries have made their emergency services available to the public through a nationwide telephone number. Calls to such an emergency number are routed to a call centre known as an *Emergency Control Centre* (ECC) where trained operators answer them and dispatch the type of emergency response team that is required. In many cases the operators use *Computer Aided Dispatch* (CAD) systems to coordinate the rerouting of police, firefighters and ambulances to the location of the emergency. CAD is a *safety-critical*, *real-time* system which should have *zero downtime*. In other words, lives could be endangered if the CAD system is unavailable, slow, fails to work correctly, or the data it contains is inaccurate.

Figure 1: A CAD system screen which is being used by a control room operator to dispatch several fire engines



[Source: adapted (cropped) image: CAD 8-12 paint.png (https://commons.wikimedia.org/wiki/File:CAD_8-12_paint.png). Image by MPD01605 under copyright (CC licence: https://creativecommons.org/licenses/by-sa/3.0/).]

Before the use of smartphones and tablets became widespread, CAD Systems communicated with police, firefighters and ambulances using mobile data terminals, two way radios and text pagers. Messages were sometimes transmitted by UHF or VHF radio because when these systems were created, wireless data communications technologies such as 3G were not widely available or were unreliable.

CAD systems were built to handle a specific maximum number of simultaneous users, which was a limit imposed by their architecture, protocols and reliance on different technologies. In older CAD systems, it is hard to increase their capacity beyond the upper limits for which they were designed, without replacing the complete system and risking failures or downtime (as happened in the case of the London Ambulance Service Computer Aided Dispatch System in 1992).

Figure 2: A mobile data terminal in a police patrol car receives instructions from a CAD system via UHF Radio

[Source: Portable data terminal, photo by DAP Technologies, originally posted to https://commons.wikimedia.org under CC licence 1.0 https://creativecommons.org/licenses/by/1.0/deed.en]

Bangbai's CAD system

The city of Bangbai has grown very rapidly over the last decade and it has become apparent that its CAD is no longer sufficient. This came to light in several stories published on social media and local newspapers, such as a police car that arrived at the wrong address to prevent a bank robbery, and two additional fire engines that were sent to fight a fire which had already been extinguished an hour earlier. Although the majority of daily cases are still handled correctly, there have been an increasing number of errors that could have cost lives. These incidents have prompted Tania Gupta, the Mayor, to ask a team led by Rahul Pandey, the Chief Technical Officer, to investigate and propose a solution.

Problems of scale

Rahul started by investigating the reported incidents and concluded that nearly all were related to the system experiencing a higher level of use than it was originally designed to handle. In a few cases a hardware component had failed and had to be replaced, but most errors were simply due to a lack of capacity to handle the increasing demands of a larger population with accidents being reported more frequently than before.

Rahul looked at several successful systems in larger cities that had coped well as the population increased. He learnt that the growth of these systems depended on them being built on a scalable and extensible architecture right from the beginning of the project. They also used common standards and protocols to make them both *future-proof* and compatible with other information systems. Successful examples included *commercial software*, cloud-based SaaS, and *custom software* developments, but all were built on a *scalable architecture* with the final choice of product/service/development depending on the requirements of each city.

An *emergency management information system* (EMIS) for Bangbai

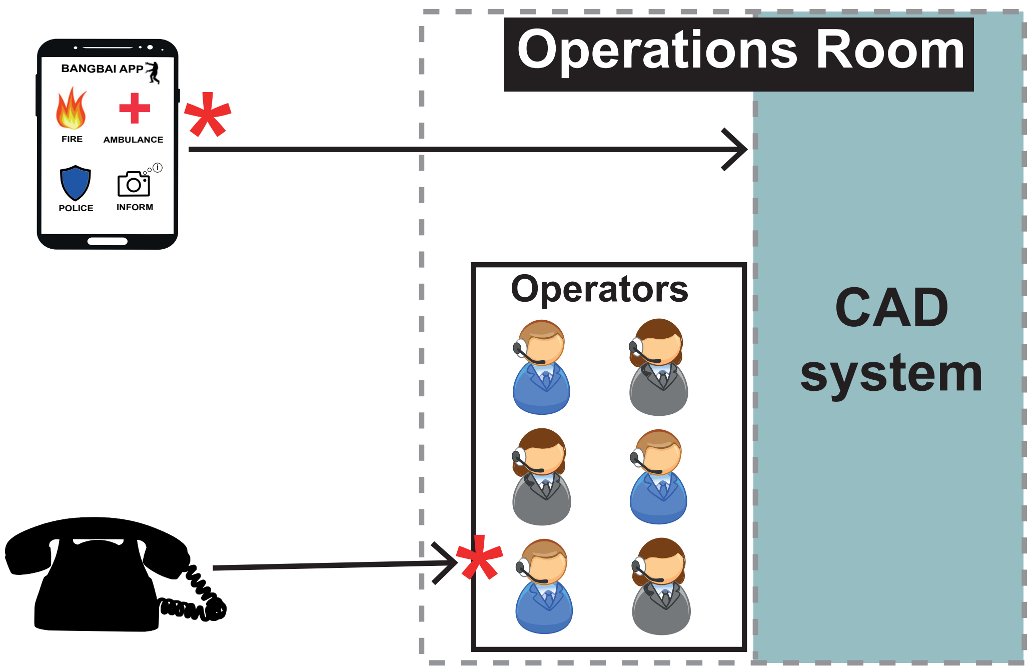
Rahul learnt that most cities were moving towards an EMIS rather than CAD. An EMIS would help the city to better manage disasters as it covers not just the response to an emergency, but also preparedness (building contingency plans beforehand), risk mitigation (noticing danger, determining potential future threats) and recovery (calculating the costs of a disaster and planning recovery, rebuilding *etc*). Rahul wanted Bangbai to eventually have a complete EMIS but understood that the first priority was to replace the existing CAD system with one that was more suitable while still allowing it to be extended in the future.

Rahul set about summarizing what he had learnt and creating the first draft of a list of requirements. He was sure that he had seen enough examples to be able to specify the necessary functionality for the Bangbai system to be a success. Rahul and his team presented their recommendations to Tania along with an estimation of the costs involved in its design, development, implementation and testing.

An emergency services app for citizens

Rahul informed Tania that the use of smartphone apps rather than telephone calls to report incidents and give information had allowed for the downsizing of the ECC in some cities, while still improving the speed and quality of data exchanged with the users. This is because in many situations, smartphone apps can gather *global positioning system* (GPS) information and providing an accurate location directly to the server with the emergency request. The accuracy of the location is critical for the success of an emergency response, as providing an incorrect address could delay the arrival of help to the victims.

Figure 3: The use of smartphone app (top) versus standard phone call (bottom) to report incidents



* Point at which location information is added to the request

[Source: © International Baccalaureate Organization 2019]

A smartphone app could also improve communications between the caller and response units by providing each with continuously updated information as the situation progresses. After the app has sent the location of the device and the type of emergency (fire, road accident *etc*) it can continue to update the server with any changes of location and notify the app user of the time remaining until the arrival of the emergency services. The ECC staff can monitor requests made using the app, and at any time they can initiate a conversation with the app user via *VoIP* to request clarification of the situation or to give instructions. The app would simplify the user experience by handling all forms of communication between the app user and the EMIS, independent of other software installed on the smartphone. Rahul advised that this may require creating a specific protocol that used more than one communications channel (for example, multiple *TCP/IP Sockets*) to simultaneously transmit different types of data. It would also require an *application programming interface* (API) so that it could use the services of the CAD system, and Rahul was told that *representational state transfer* (REST) was the best way to do this.

Users without the smartphone app installed would need to phone the emergency number, and ECC staff would ask for the caller's location details before submitting the request via their desktop application. There was a sufficient number of operators to cope with the current volume of calls, and Rahul's team of experts predicted that the use of the app would drastically reduce future demands on the call centre operators. Therefore, Rahul decided to monitor usage of the emergency number and reassign ECC staff to other roles if the app resulted in fewer phone calls. Meanwhile, Rahul cancelled any further investigations into the feasibility of upgrading the existing call centre technology.

Reporting / informing about non-urgent issues

The emergency app would also provide an opportunity for the risk-mitigation element of an EMIS, *ie* pre-empting future problems and acting before they become serious. In addition to requesting emergency services, the smartphone app would make it possible for citizens to report non-urgent issues, such as traffic congestion, suspicious activities, risks to public health and safety, or even suggest improvements to services.

Figure 4: An example of a poster promoting the use of the "Inform" function of the Bangbai App

[Source: traffic image from Max Pixel]

When used in “Inform” mode, the app would upload a photo and a text description or a recorded sound file, along with the location of the non-urgent issue (which might be different from the current location of the phone). The server would also receive additional data about the device which may include information that personally identifies the citizen who owns it. Some potential users have asked for location services to be turned off, or would access the EMIS using a *virtual private network* (VPN) or *proxy server*.

Emergency response vehicles

Rahul continued to explain that as soon as an emergency response request is received by the new system, it would search for the closest emergency response team which is able to deal with the requirements of that situation. This team would then be automatically dispatched to the required location.

The existing CAD system already has a large number of dedicated devices with an embedded operating system, installed in most emergency response vehicles. Until now these legacy devices have been used to update the central server at regular intervals with their location in a stateless way. However, since these devices use *HTTP* as a protocol, Rahul decided that they could be used to provide more functionality to the emergency responders. This might include messaging systems and queries which would require *stateful* communication using either *cookies* or *URL rewriting* to maintain the state between requests.

***Multitier architecture* and scalability**

To handle the increasing workload produced by citizens reporting many non-urgent issues and requesting the automatic dispatch of emergency services, the new system needs to be built on a scalable architecture and be compatible with future developments in other areas of EMIS. Rahul’s team of experts explained that best practice was to separate the system into tiers which could then be distributed across as many servers as necessary to meet the demands of the users. He knew that this would allow the main components to be reusable and, for example, a web interface for reporting emergencies and requesting assistance could be added with very little additional effort, as it would simply use the same logic tier services as the smartphone and desktop apps.

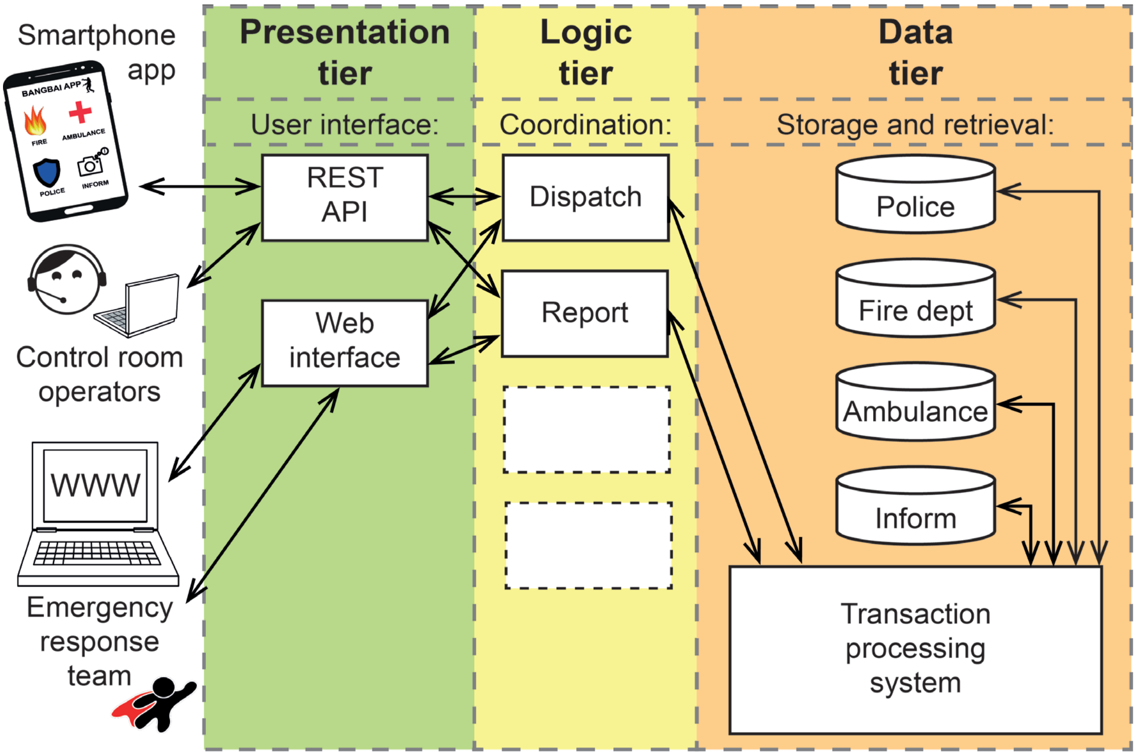
The new CAD system (the ”Response” part of the Bangbai’s new EMIS system) will be implemented on a *cluster of servers* which together answer queries and provide the services that users need. The different parts of the system will be divided into tiers using a “Multitier” architecture so that in future, these can be spread out across many different servers to increase capacity. The three main tiers will be called “Presentation”, “Logic” and “Data”.

The Presentation Tier will contain the code for the user interface, which is the part of the system that communicates directly with the smartphone app, ECC staff and the devices used by teams of emergency responders.

The Logic Tier has algorithms that contain the main procedures/services of the EMIS, for example, all of the operations needed to coordinate the response vehicles.

The Data Tier encapsulates the code which accesses the databases or other data sources. In such a complex system, the data may be spread across many different types of database which will be hosted in different physical locations. All changes to the data should be carried out inside transactions and controlled by a *transaction processing system* (TPS) which means that if there is an error which prevents any part of the operation from completing, all the data will be rolled back to its original state before the operation began. Alternatively, if all the parts of a transaction succeed then all the included changes in the different databases will be saved (committed).

Figure 5: The multitier architecture



[Source: © International Baccalaureate Organization 2019]

In multitier architecture programs, a tier should only communicate with other programs in the same tier and/or the tiers next to it. For example, a client-side app should be prevented from requesting information directly from the Data Tier without first passing its request through the Logic Tier which would apply the appropriate security and credentials checking, as well as making sure that everything else is in order before allowing it to proceed to the Data Tier via the TPS.

Load balancing algorithms

Rahul is concerned that as Bangbai is growing so rapidly, the increase in usage may generate excessive load on the servers at very busy periods. He identified several load balancing algorithms that can be used to address this problem by redirecting requests to any one of a group (*cluster*) of server machines that would be capable of handling the request.

When more servers are added to a system they do not necessarily all perform exactly the same functions or services. For example, if one particular service is very heavily used, then more servers could be added to handle that particular type of request. The distribution of concurrent users across these servers might take into account which services the clients are trying to access as well as other factors such as the current workload of the different servers that are available to respond to that type of enquiry/service.

Rahul has been investigating the following algorithms for load balancing:
- *client side random*
- *weighted round robin*
- *source IP hash*.

Rahul is also concerned that many citizens of Bangbai will use the reporting function of the app to upload relatively trivial information such as images of trash on the sidewalk, cars with flat tyres and stray cats *etc*.

Failover

If a server becomes unavailable for some reason during normal operations, its functions are taken over by another server which is capable of continuing its work. In some cases, the second server will need to have details of the current state of interactions with the users of the first server so that it is able to continue dealing with these users seamlessly.

Reliability

Tania responded positively to Rahul's ideas, however she asked him to ensure that the new EMIS system would be as reliable as possible by:

(a) allowing the capacity of the system to be expanded easily (*ie* scalability) as required by adding new servers;

(b) incorporating duplicate (*ie* redundant) hardware which is ready to immediately take the role of the currently used hardware should it malfunction (*ie* failover).

Challenges Faced

There are a number of challenges that need to be addressed in the development of the Bangbai EMIS. These are to:

- begin the planning of an EMIS system which handles all the current CAD work in a more scalable and extensible manner.
- plan for the increasing loads on servers by investigating proven methods, including providing *redundancy*, failover and load balancing.
- develop a smartphone app that has emergency functionality and a reporting / informing function that allows citizens to provide information on non-urgent issues.
- make use of the legacy devices already present in the emergency response vehicles while still meeting the requirements of the proposed system.
- report on the implications of the project with respect to the key stakeholders including an overview of any obvious legal and ethical implications.

Additional terminology to the guide

Application programming interface (API)
Client side random
Cluster
Cluster of servers
Commercial software
Computer aided dispatch (CAD)
Cookies
Custom software
Emergency control centre (ECC)
Emergency management information system (EMIS)
Emergency number
Failover
Future-proof
Global positioning system (GPS)
HTTP or HTTP/2
Load balancing algorithm
Multitier architecture
Proxy server
Real-time
Redundancy
Representational state transfer (REST)
Scalability / scalable architecture
Safety-critical
Session
Session IP hash
Session management
Socket
Source IP hash
Stateful / stateless / maintaining state
TCP/IP sockets
Transaction processing system (TPS)
URL rewriting
Virtual private network (VPN)
Voice over internet protocol (VoIP)
Weighted round robin
Zero downtime

Some companies, products, or individuals named in this case study are fictitious and any similarities with actual entities are purely coincidental.