Urgent Ambulance Help Application Concept

Author: Dmitrii Egorov

Date: 15/08/2021

Content

About the Author

My name is Dmitrii and I'm a software architect. I have worked in the IT domain for more than 10 years. Currently I live and work in London in Deutsche Bank as Java Fullstack Developer. In this document I propose an idea that can improve NHS services. I spent a year investigating what technical design might fit this system. I even started to create it from scratch but at some point I understood that I have no enough time to do it (I have only evenings after my working day).

This work is just volunteer work and I don’t sell anything here or represent someone’s interests. All technologies I used in technical design are free and open-source. I have lived in this country for only a couple of years, but I have already noticed the worthy work of ambulance specialists and I would like them to improve their IT infrastructure as much as I can.

If you need to contact me my details are here:

Dmitrii Egorov

Phone: 07495070530

Address: Flat 49, NW10 3TJ, Donnington court

Email: isicju@gmail.com

Introduction

This document describes the theoretical and technical design of an application that can significantly improve ambulance service. The main idea is pretty straightforward – create a mobile application that can stream video and audio to ambulance officers.

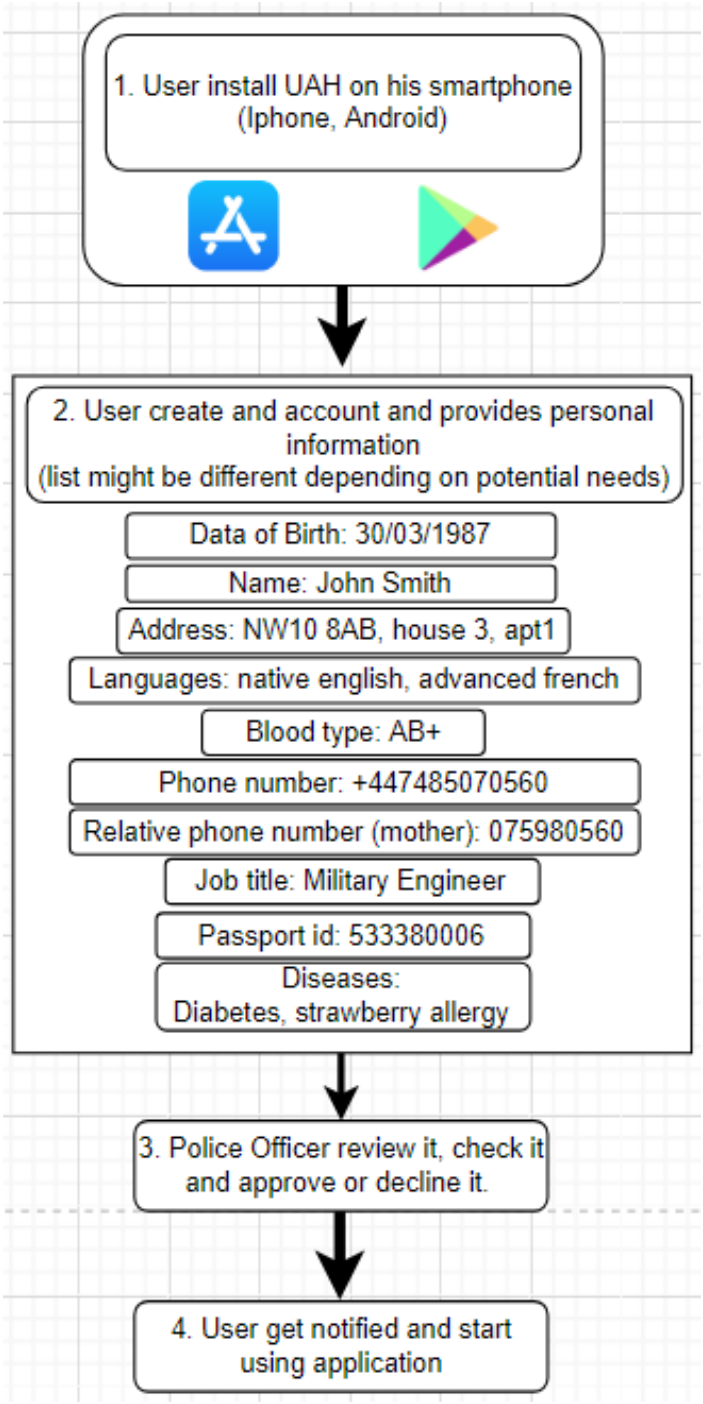
Such systems might have different technical designs depending on functional requirements. In this example I'll provide one possible architecture among many possible. From my perspective it's more important to share ideas and the fact that those ideas can be implemented without spending an enormous amount of time and money. At the same time these ideas can save humans lives and improve health services.

Main Scenario

Here I describe the main user's journey and the most common use case of using UAH application.

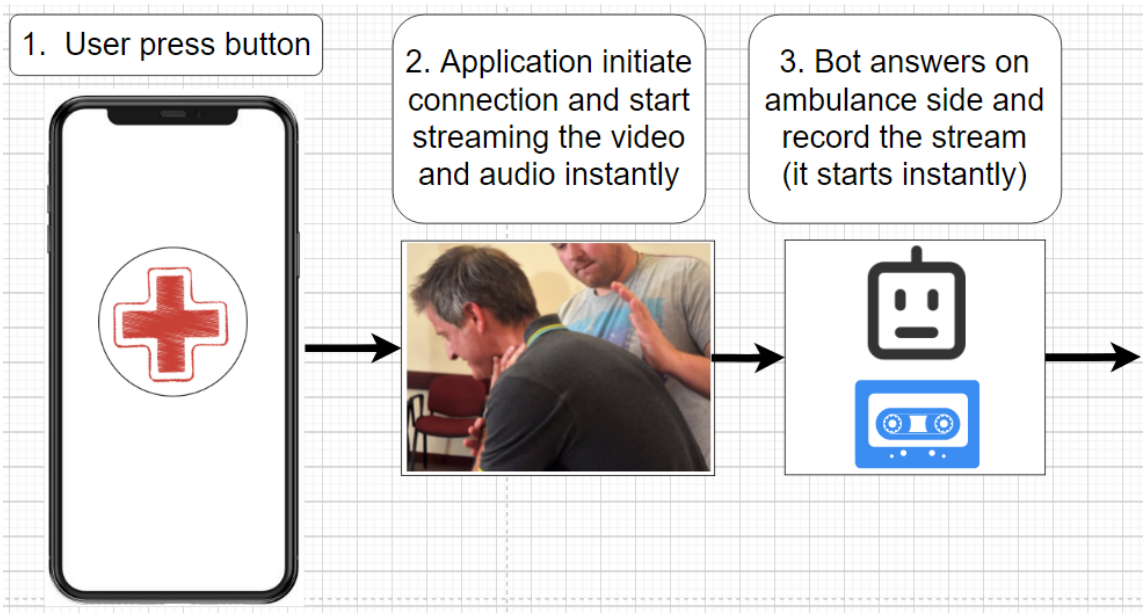
Step 1. Installation and registration survey

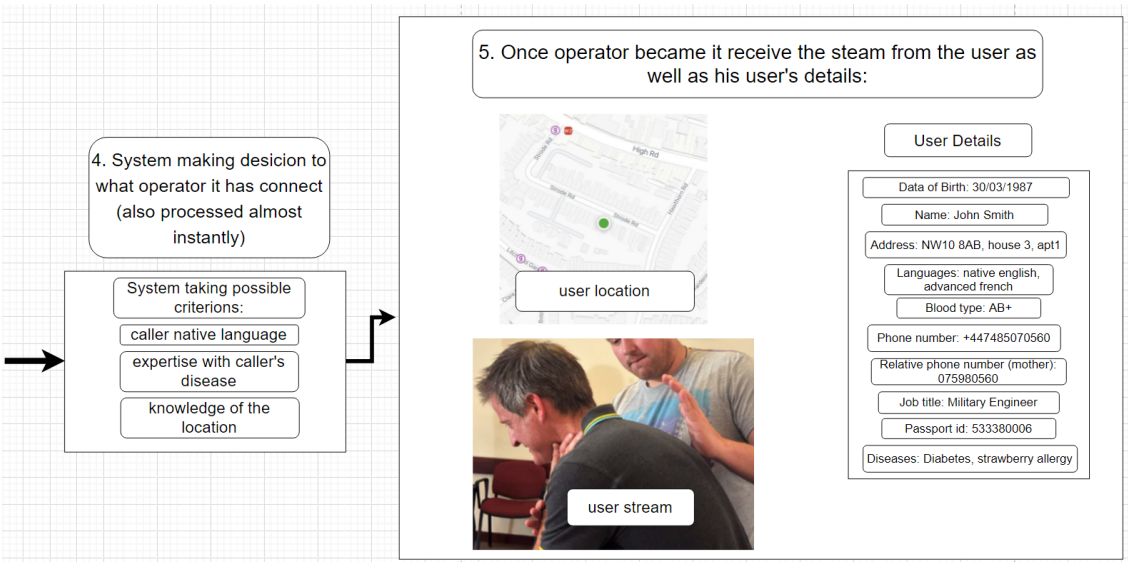
Before using the application all users have to pass registration steps and provide information required for validation and for future needs (for example user's age, speaking languages, etc).

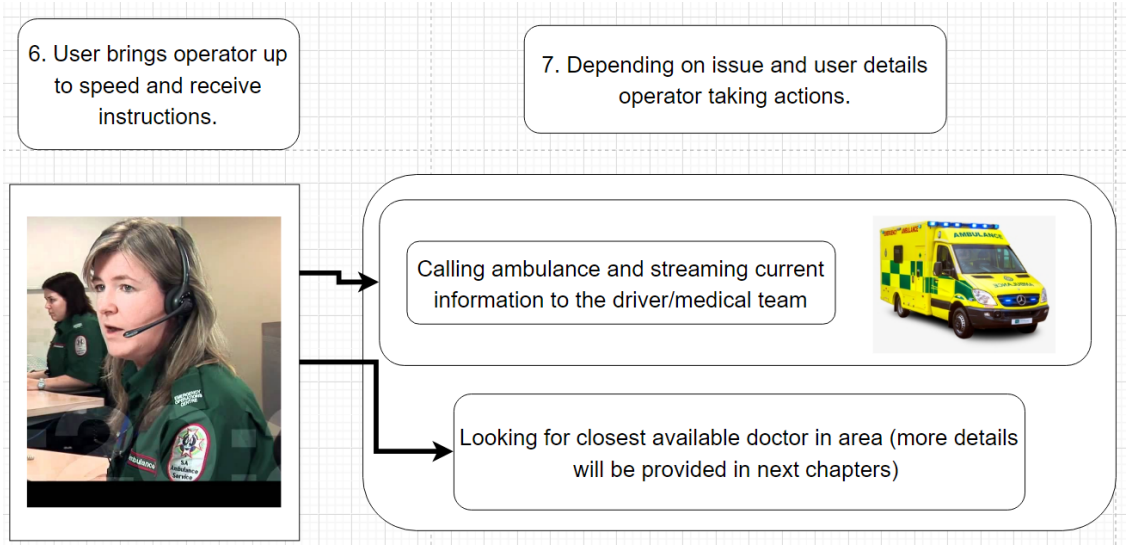


Step 2. Using application in case of emergency

In case of emergency user press button and application immediately starts streaming video and audio from his camera and microphone:



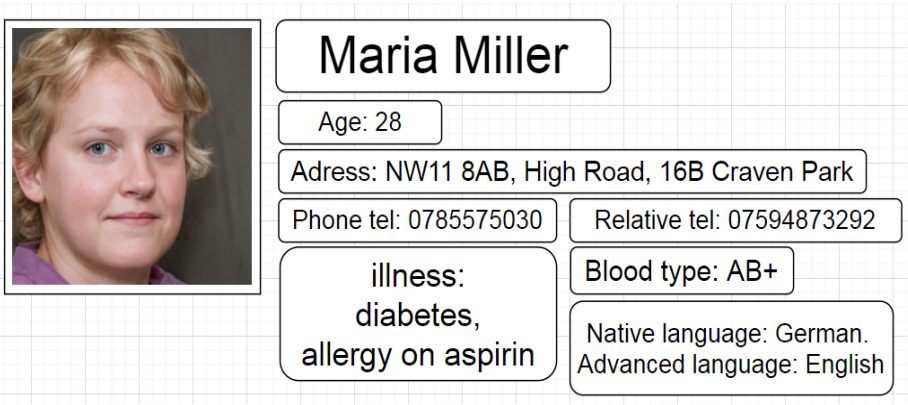




System advantages

Such system especially at advanced levels, can significantly improve ambulance services. Let's take a look into the next example. Once connection is initiated operator might be see next information:

What operator might know about the caller:



What operator can see from the caller video stream:

Application's advantages list:

- System will finds operator with the most "compatible language" depending on language caller speaks including finger language
- For hearing impaired operator might use language of the deaf or simple show text on the screen
- Having video and audio stream operator might immediately understand the problem even caller can't explain situation properly (caller might be in shock or not just capable to formulate the problem)
- Operator might share video/audio stream with ambulance team in order let them be prepared when they come
- Having list of caller diseases operator might try to guess what might be the reason of the call (low/high sugar in blood) or allergy on aspirin taken by mistake
- With video connection operator might check that given instructions are done correctly by caller and guide him during the call

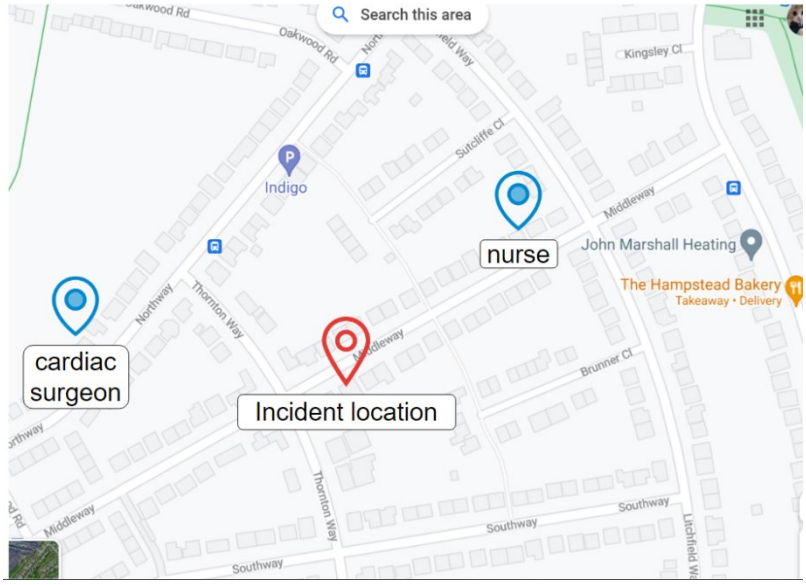
Summarizing – system will reduce time required for operator to make a decision. I expect in many cases it might be a matter of life and death.

More system features.

In the previous chapter we reviewed a simple case when a user calls the emergency. But such systems might have different functionalities. In that part we review potentially useful functionalities.

The possibility of an operator to connect with specific user

Such system also might easily provide connectivity from operator to client. We can expect that such applications will be used by many citizens including medical specialists. Operator might call a specialist nearby and ask him for help. In the case of big traffic when ambulances just can't come within a short time (COVID-19 pandemic was a good example when most ambulances were busy) having specialists might be very helpful.

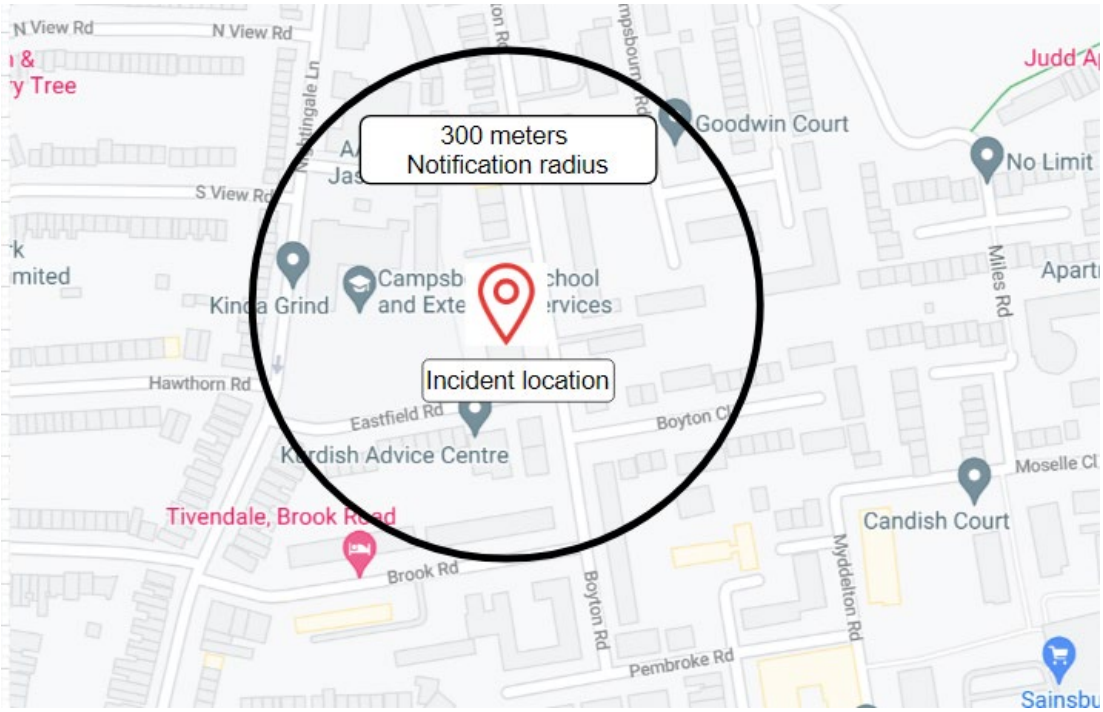


Obviously such users have to accept such call from ambulance operator when inside their installed application configuration.

The possibility of an operator to notify many users in area

Ability to connect and notify clients about potential threats could be also useful. For example:

- Citizen returned from Safari in Africa and he was diagnosed plague virus then all citizen that live near his house will be notified to stay home
- It was discovered that in some local store some vegetables been infected then all citizen near will be notified to drop vegetables if they purchased it in that store
- Water pipes were polluted by chemicals after the factory explosion and tap water became poisoned. In such cases, quick notification might help citizens to avoid intoxication.



Unique Id on Phone Screen

In large number of situations caller make call in order to help someone else. In such situation all potential information about caller will be useless. But if person that need for medical help will have such application installed on his phone then operator might easily recognize him using unique id feature.

The idea is to generate id for each user that install that application and keep that id on his phone lock screen. In that case even without pass code caller can find id of sick person and tell it to the operator. There is no need to explain how important is to know information about disease/blood type/etc might useful.

There is no private information violation. Id even exposed to public wont expose any personal information. Only operator will be able to use. It's enough to have 6 characters long length (a-z) to cover 300 million citizens.

Quadcopters quick delivery - Near Future technologies

Implementation of ideas described earlier have no technical blockers and some applications already provide such functionality (e.g. zoom/skype for video connection, maps – google map, etc). But the feature proposed in this part is not yet implemented somewhere but still pretty doable.

Quadcopters in ambulance service

The main purpose of quadcopters is to deliver everything required for first aid. Quadcopter can bring first aid kit within minutes or even seconds and might be even save life. Equipment from first aid kit can be instantly used by even not educated person.

Here I provide list of examples when instant reaction might save caller life:

Situation	What Quadrocopter can bring and how caller can save injured person.
A person has lost a limb after being hit by a train or his limb is crushed by a wall. Or just with bleeding (eg bullet or knife wound)	Quadrocopter brings tourniquet and anesthetic syringe. Caller makes an injection and applies a tourniquet to prevent bleeding.
A person was bitten by a snake or bitten by a wasp and allergic to it	Quadrocopter brings a syringe with an antidote. Caller makes an injection.
A person has low blood sugar	Quadrocopter brings blood sugar tester and two syringes for lowering or raising sugar in blood. Caller uses tester and after make injection depending on sugar level in blood.
The person has been diagnosed with cardiac arrest or he is not breathing	Quadrocopter brings heart rate monitor and defibrillator with oxygen pillow. Caller verify that there is no heart beat uses oxygen pillow and defibrillator (following instructions from operator)
The person choked and the food got stuck in the throat.	Quadrocopter brings surgical knife and tube for Tracheotomy. Caller perform Tracheotomy if there were no other options (also with operator instructions)

A person got many burns from a fire.	Quadrocopter brings burn ointment and anesthetic syringe. Caller uses inject anesthetic and uses burn ointment covering damaged skin.
Many other scenarios.	***Quadrocopter bring something that is not too much heavy. And caller saves injured life.***

As far as I have no education might be some examples I've provided are naïf but I believe in many situations quick actions might save lives. Begging from simple situation with pain shock up to blood loss can be avoided with simple equipment and with person without medical education (but guided by operator).

Number of Quadrocopter required to cover central London (30 km radius)

Quadrocopter can be much faster than ambulance vehicle especially in case of traffic or pandemic (when most of vehicles are busy). Quadrocopter has higher speed than ambulance as well as shorter path to caller. If we assume that Quadrocopter has to arrive within 60 seconds then for London area we would need:

1. Radius of Central London: 30 KM
2. Minimum Quadrocopter speed: 80KM/H
3. ETA: 60 SEC

Distance that Quadrocopter can reach within 1 min: 1,3333(3) KM

N of quadrocopters = 2*PI*(R London)^2 / 2*PI (R Quadrocopter)^2. = 30000^2/(1333^2) ~ 506.

So roughly 500 Quadrocopters might cover the central and provide help with about 60 seconds ETA. But we didn't consider time required charging Quadrocopter with correct equipment but I expect it can be done quickly. I can't provide potential price estimation because there is no price on the market for such devices. I expect the price for such device might be about 5k-10k pound for each device so in total only Quadrocopter cost would be 25M pounds. Provided numbers and costs might just give indicative values and generally are not reliable.

Technical Application Design

This part would be rather interesting to the technical team than managers. Here I provide possible technical design and requirements for such application (part with Quadrocopter not on the list). So the core functionality for this app is shown in “Main scenario chapter”. The key objective of this part is to provide architecture and estimate how many efforts (man-days) are required to create it. Having such information we can estimate how much time and money are needed to implement the system.

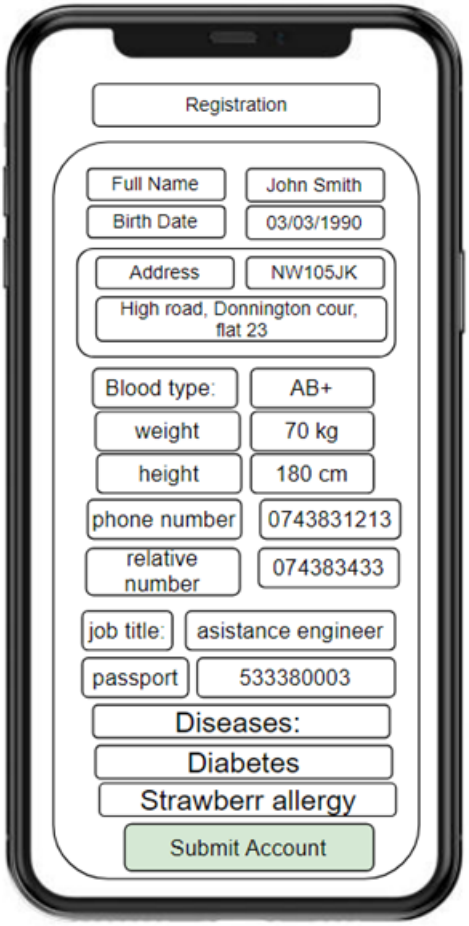
Further calculation is not precise but I can give a rough estimation in man days (MD).

Client Part – Functionality on Smartphone side

Estimation for client part will be doubled as far as application has to support IOS and Android operation systems

Registration Screen.

We assume that registration survey won’t be too big.

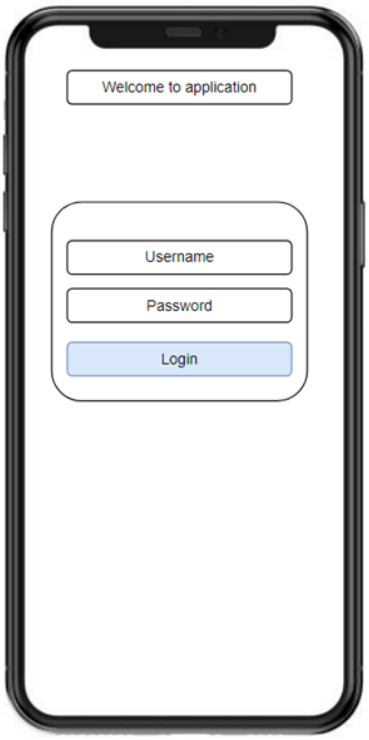


registration screen	26 UI elements	5MD
network	1 request	3MD
total		8MD
total for IOS and Android		16MD

Considering only the main screen without exception screen or confirmations screen it might take about 16 man days for registration screen. (I missed part with profile image but it won’t increase the size of the task too much).

Login Screen

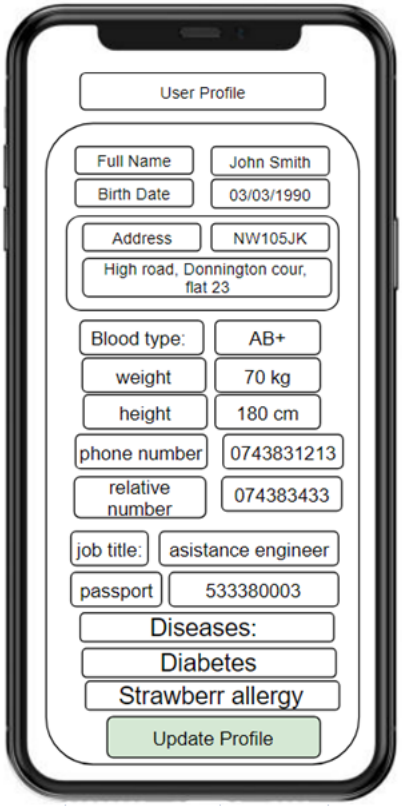
Once account is approved user can be given login credentials. Login screen is straightforward, I estimated it to 8 man days.



login screen	4 UI elements	1MD
network	1 request	3MD
total		4MD
total for IOS and Android		8MD

User Profile Screen

Now when user is logged in he can adjust his profile data. Such screen is almost the same as a registration screen. So we also consider 16 man days estimation.



user profile screen	26 UI elements	5MD
network	1 request	3MD
total		8MD
total for IOS and Android		16MD

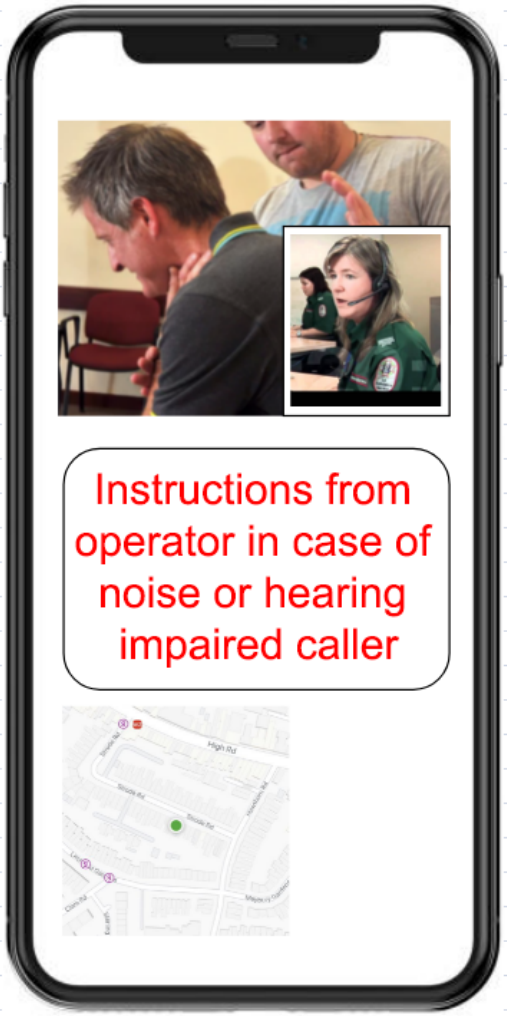
Call Emergency Screen

Such screen can be more featured but in this example let's simplify it. There is nothing but one simple button to press during the emergency.

login screen	1 UI elements	1MD
total		1MD
total for IOS and Android		2MD

Streaming Screen (The core screen)

This screen is the most important part of the application and will be shown to the client one connection with the operator is initiated. From an implementation perspective the heaviest part here is hidden from the user.



Outdoing video screen	1MD
Incoming video screen	1MD
Instructons screen	1MD
Implementing WebRTC connection with server	60MD
Map screen	1MD
Sharing Location address	5MD
Total	69MD
Total for IOS and ANDROID	138MD

So rough estimation is 138 MD. It's pretty difficult to estimate properly because functionality itself is big. Also we have to consider that WebRTC library (protocol responsible for video and audio) might have different implementation in IOS and Android environments. More details about WebRTC protocol I'll provide in chapter with backend estimation.

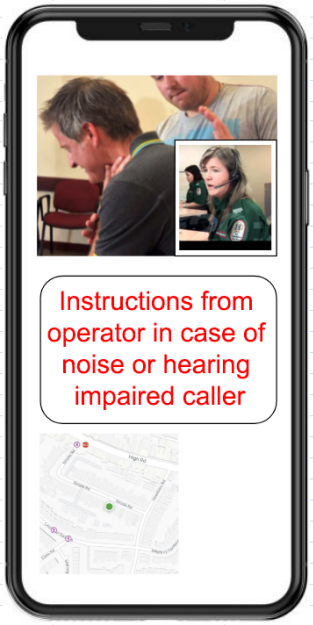
Notification Screen (The core screen)

In example earlier I shown 2 possible scenarios when operator might:

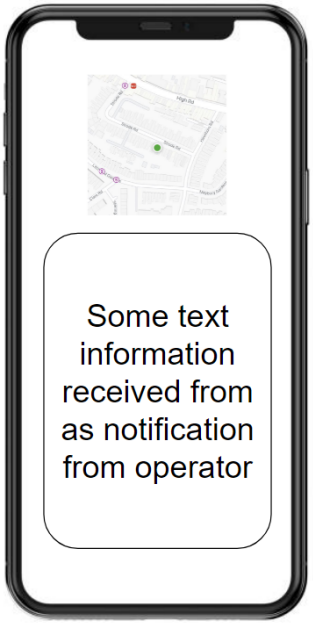
- contact closest medical specialist and ask for assistance
- notify all users in an area.

For both cases we need to 2 screen:

- connection with operator (we currently have similar screen so we assume we can reuse previous screen)
- Screen with simple notification from operator is pretty straightforward



Reusing existent screen	1MD
Total	1MD
Total for IOS and Android	2MD



2 UI elements	1MD
Listening server for notications	5MD
Total	6MD
Total for IOS and Android	12MD

Both screens together are estimated to 14 MD.

Unique Id on Phone Lock Screen

1 UI element on screen	5MD
Network call for fetching user id	5MD
Total	10MD
Total for Android and IOS	20MD

Total Estimation On Client Side

Total estimation for all mobile screens considering 2 operation systems is 214 man days. But such work would require 2 specialists – Android and IOS developers. Breakdown for such screen is:

Registration	16
Login	8
User Profile	16
Call Screen	2
Streaming Screen	138
Notification Screen	14
Unique ID on Lock Screen	20
	214

Shown screen might miss some additional elements (like exit button or labels) but in general they won't change estimation significantly. As I already mentioned, the purpose of this estimate is to find the cost range and not the exact value.

Operator Part – Functionality on Operator Screen

In this part I describe and estimate screens required for ambulance operators. We assume that the operator will use a desktop PC. But this part still doesn't include backend architecture which is the heaviest part in the system.

Operator Login Screen

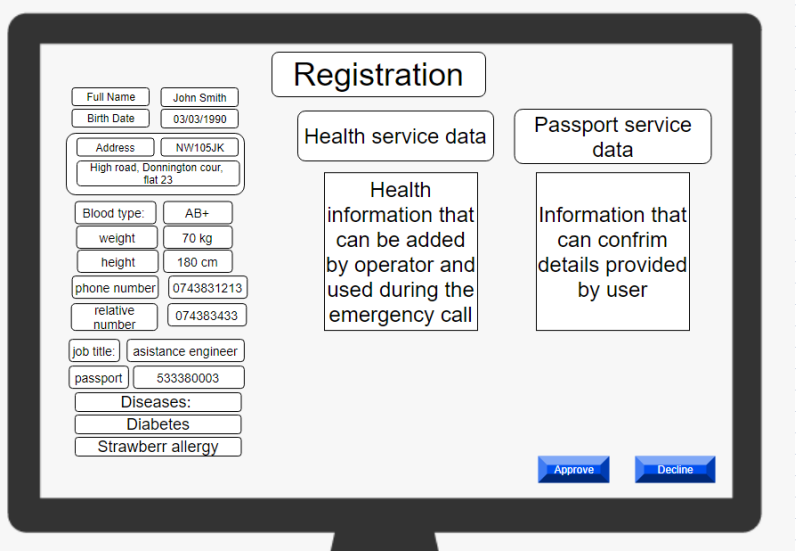
Screen for operator is pretty simple but I expect that simple name/password won't be enough and added some time needed to support security tokens (can't be usb stick with or just an application on employee's phone. More details here: https://en.wikipedia.org/wiki/Security_token)

login screen	3 UI elements	1MD
network	1 request	3MD
security token support		10MD
total		14MD

So at least 14MD are required for login screen.

Operator User approval screen

Each new user requested for a new account has to be approved or declined by the operator. This screen might be pretty complex and might aggregate different external services like data from passport center or health service registry.

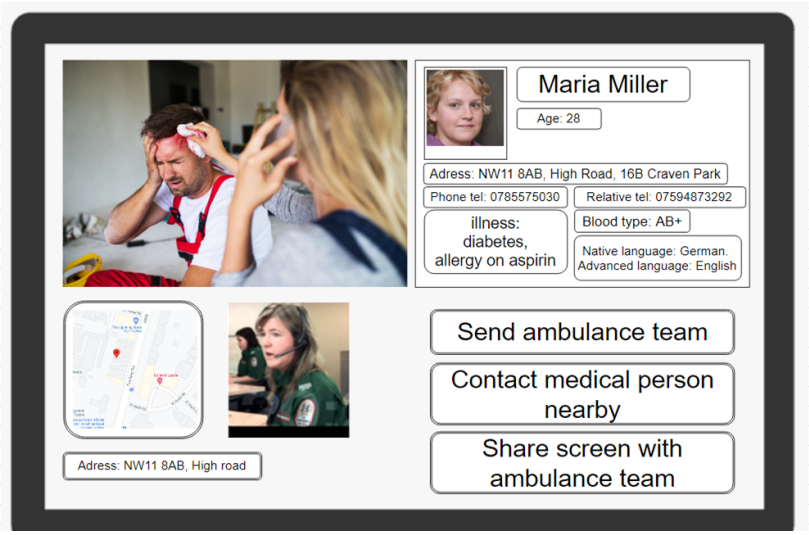


login screen	33 UI elements	5MD
retriving data from database		10MD
fetching data from health service		10MD
fetching data from passport service		10MD
total		35MD

Such screen might be much more complex. I assume that the operator might find and add much more important information that can be used during the call. For this very case I estimated work for 35 man days.

Operator Emergency main screen

Emergency screen is the core screen which is used during the call. Its complexity might be very high and it's pretty easy to underestimate it. This screen will correspond to client's phone screen I described earlier.

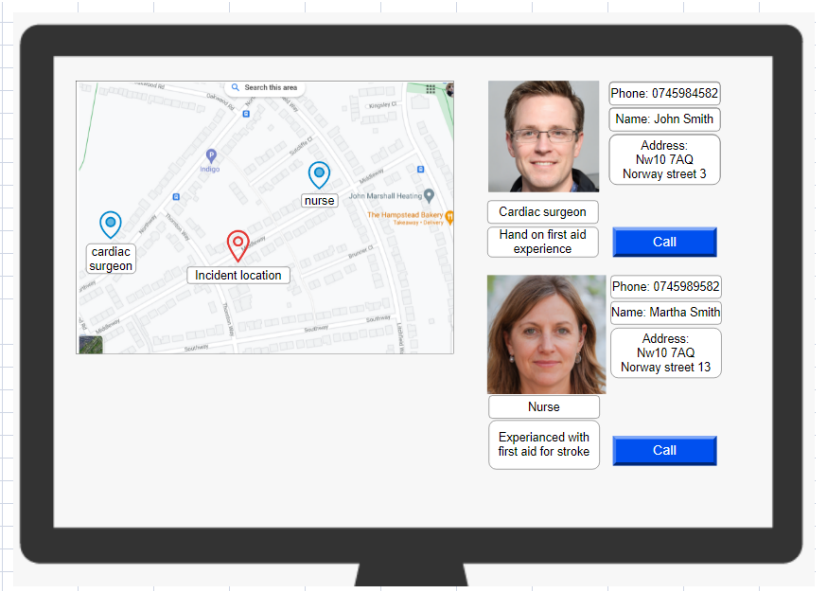


16 UI Elements	3MD
Fetching user details data from database	10MD
Implementing WebRTC connection with server	60MD
Showing mapping location	5MD
Call ambulance team service integration	10MD
Contact medical person nearby service integration	10MD
Share Screen with ambulance team integration	20MD
Total	118MD

For all screens and integrations application requires at least 118 man days.

Operator Contact medical specialist nearby screen

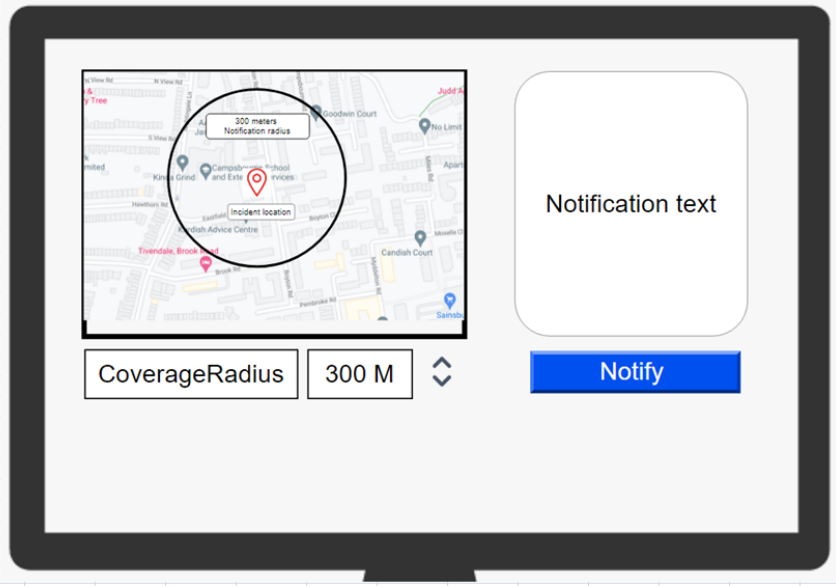
On this screen operator might contact with medical specialist and ask him for help while ambulance vehicle is coming.



16 UI Elements	5MD
Fetching data about specialist nearby	10MD
Showing address on the map	5MD
Call user functionality	10MD
Total	30MD

Operator Sends notifications in area screen

This Screen allows operator notify all users in area with given radius.



6 UI Elements	2MD
Fetching data about users in given area	10MD
Showing address on the map	5MD
Sending notification to users	10MD
Total	27MD

Notification screen is pretty simple and I think 27 man days is pretty reliable estimation.

Total operator screens estimation

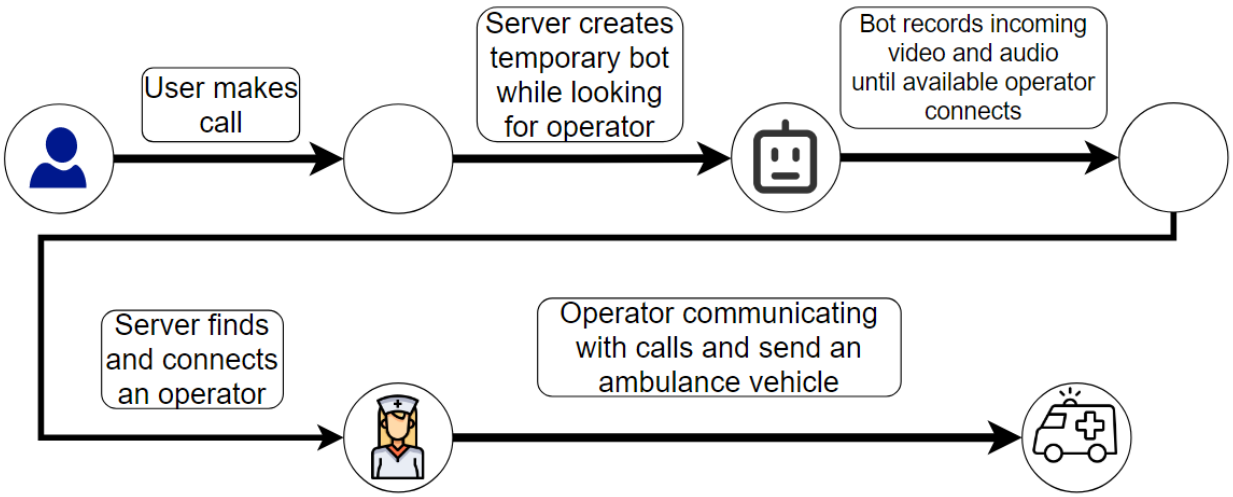
In total 5 operator screens require about 224 man days.

Operator login screen	14
Registration new client screen	35
Operator emergency main screen	118
Operator calls medical person nearby	30
Operator sending notifications in area screen	27
	224

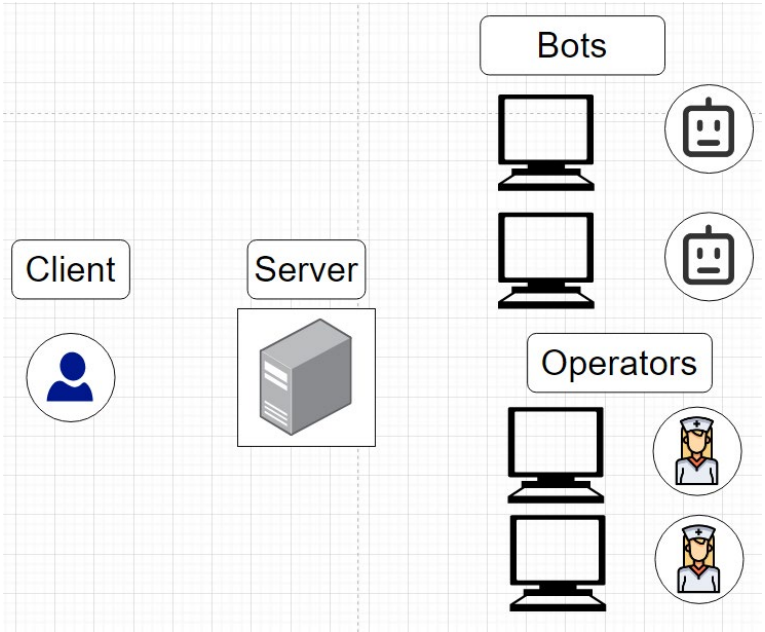
Server Part – Architecture and it's estimation

This part is the most complex and its implementation will affect development on client (phone) and operator sides. Here I'll describe one possible solution but it's just one among many potential designs.

Before we start I repeat here the process flow:



For such process we might have next architecture:



Server has to provide many services like databases, mapping services, user details registry, calling services, etc. But the crucial server function here is to create video and audio connection between client and operator.

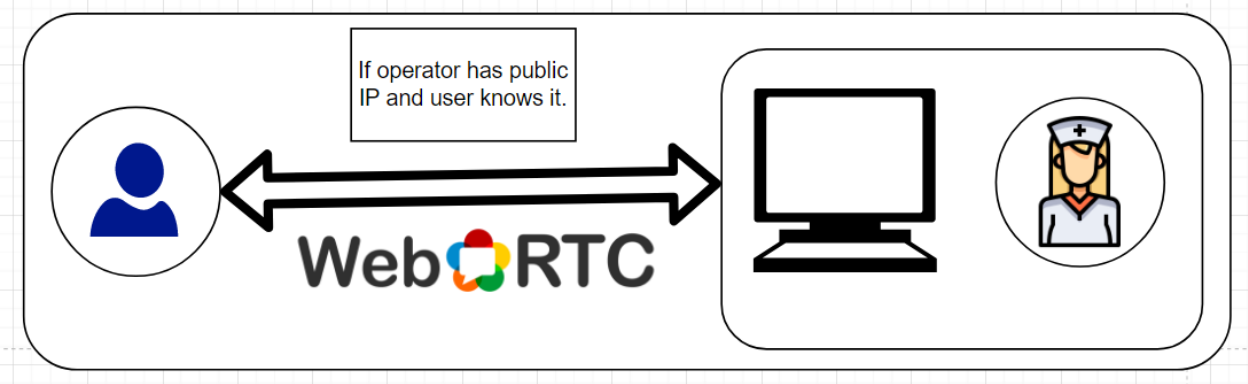
Video and Audio Protocol – WebRTC

The main protocol in such system should provide stable and quality video and audio connection. Such protocol has to be supported by Desktop (Windows or Unix) , Android and IOS. WebRTC (web real time communication) is free and open source project that provides a platform for video audio communication for all mentioned operation systems.

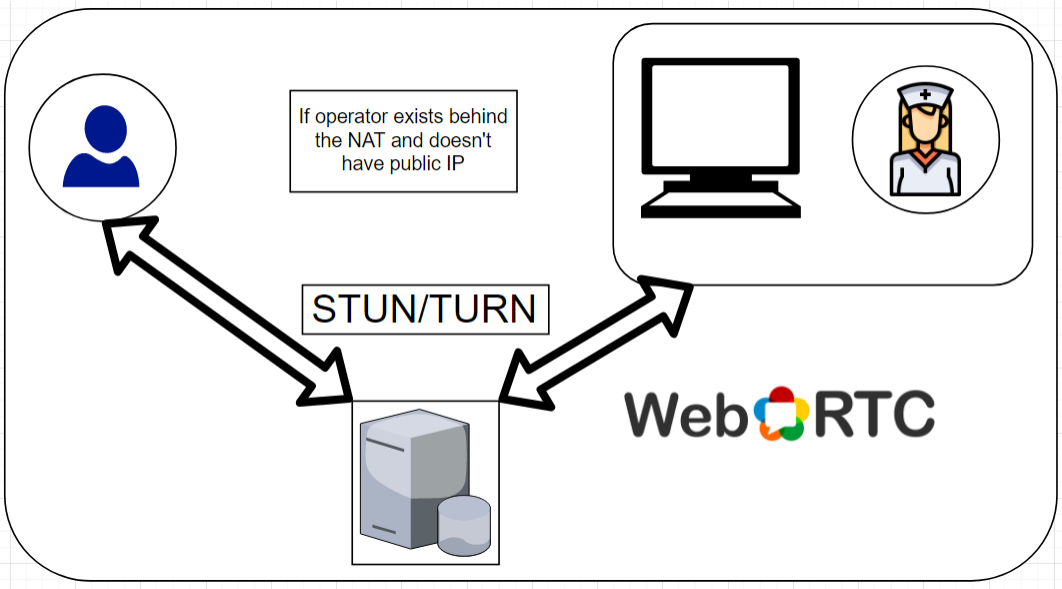
Again – video and audio protocol in such systems is the most important part. Some software architects might even recommend creating a custom protocol on top of TCP/IP. But so far WebRTC protocol has proven it's reliability and now it's widely used in the enterprise world (for example Google hangouts, Discord, Facebook messenger, Amazon chime, etc) and seems to work fine.

How WebRTC arrange communication

WebRTC can support UDP and TCP protocols and it can dynamically lookup for the best connection between current user and operator. WebRTC communication might exist in different network environments, the easiest case when user knows operators IP:



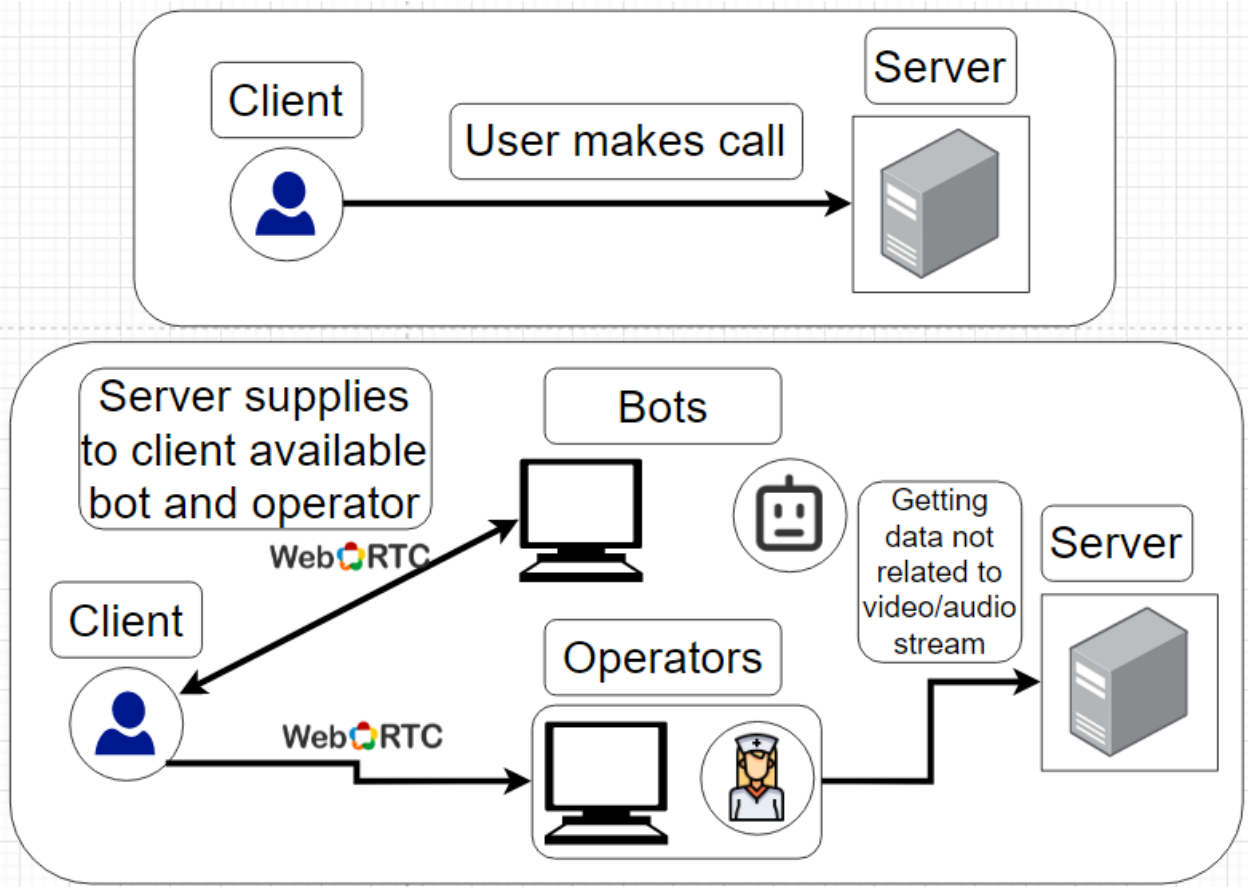
If an operator exists behind the NAT and the client can't make a direct call then we have to configure the STUN and TURN server. STUN and TURN server will play a role of intermediate node. User and operator can use it because they both know the STUN/TURN server IP address.



More details and examples can be found here: https://webrtc.github.io/samples/ . There are many WebRTC aspects and it provides many solutions from the box like dynamic video quality or conference calls.

Server Architecture with WebRTC

In this design I assume that operators have public IP (just to simplify the architecture). To describe WebRTC integration I show here 2 schemas. First schema shows that the client knows only server IP and calls it. Right after the server connects the client to the bot and after connecting the available operator.



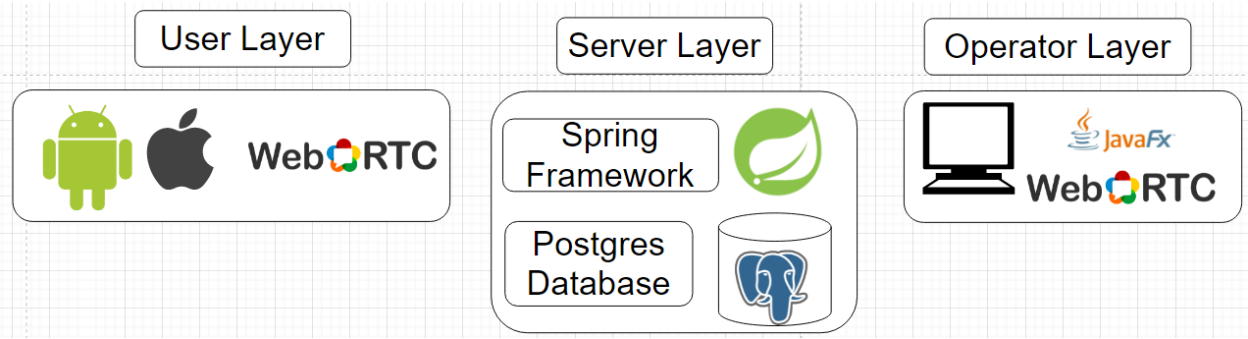
Remaining Functionality Required for the System.

Video and Audio Connectivity is not the only part of such system. Here I provide services that server has to provide.

	Solution
Persistence service	
Saving Users, Operators data	SQL server
Saving binary data	Any file system
Connectivity service	
Connection between user/server/operator	HTTP/WS protocols
Connection with external services like user documents check	HTTP/WS protocols
Maps services	
Allows operator/user see the map data	OpenLayers framework

Backend Design and Estimation with concrete frameworks

So finally considering services I mentioned in previous chapter we might have next architecture:



It might look a bit poor but I don't want to create a detailed map specifying this or that small framework. Here I provide only critical technologies. And personally I think that only video/audio streaming protocol matters here and all remaining frameworks are not that important. All frameworks and solutions are free and some of them open source.

Here I estimate all functionality mentioned on the server side. In total it takes 290 man days.

Connecting user to bot or to operator	40MD
Handling webrtc stream from bot side	30MD
Handling WebRTC stream on operator side (performed not on server but operator machine)	0MD
Saving User Details/Operator Data in database	40MD
Saving binary user details/operator data	20MD
Exposing API for User Details/Operator Data	40MD
Providing API to notify users by operator request	20MD
Providing maps API	40MD
Getting data from heath service used during registration	20MD
Exposing data from healh service to operators	10MD
Getting data from passport service used during registration	20MD
Exposing data from passport service to operators	10MD
Total	290MD

Final Estimations and Calculation of Expenses

The whole system consists of 3 parts: Clients with smartphones, Operators with desktops and Server. In previous chapters I made rough estimations and here we calculate how much effort we need. I didn't count DevOps part: arranging code versioning, deployment process, etc. I did it for one simple reason – I expect that this project can be created in an existing ecosystem which might already have this process.

Client side Android	112
Client side IOS	112
Operator side (desktop)	224
Server side	290
	738

Human Resources Expenses

If we assume that for each part we hire a dedicated developer then we would need about 4 developers. Considering calculated days and average London salary (taken from checksalary.co.uk) we would need at least 200.000 pounds. I personally expect a higher number and 200k I recognize as an underestimated value but at least it gives a rough salary estimation.

	working days	month (22 working day in average)	Average Year Salary (according checksalary.co.uk)	Cost
IOS Developer	112	5,09	71000	30121,21
Android Developer	112	5,09	68000	28848,48
Java Developer for Desktop	224	10,18	63000	53454,55
Java Fullstack Developer fo Server	290	13,18	79000	86780,30
Total cost in pounds				199204,55

Expenses related to hardware

Expenses for hardware can be split in 4 parts: Server price, Operator Desktop price, Hard Drives pool, Internet provider cost. The greatest load falls on the interaction between the client and the operator. Desktop PC should have a good CPU and at least 16GB RAM. We don't need a super performant server but for stability we need 4 nodes. And as far as video connection creates high load on the network we need at least 500MB speed. And a lot of hard drives we need in order to save all video and audio calls – in example I'll use 4 disk 14TB each (it's allows to save 112K incidents 0.5Gb each record)

After quick searching I found next acceptable options:

Server: Xeon 12 cores 2.4G, 16GB RAM. Price – 2200 pounds.

Compare

Smart Value PowerEdge R440 Server
Excellent Density and Flexibility

Designed for dense, scale-out computing

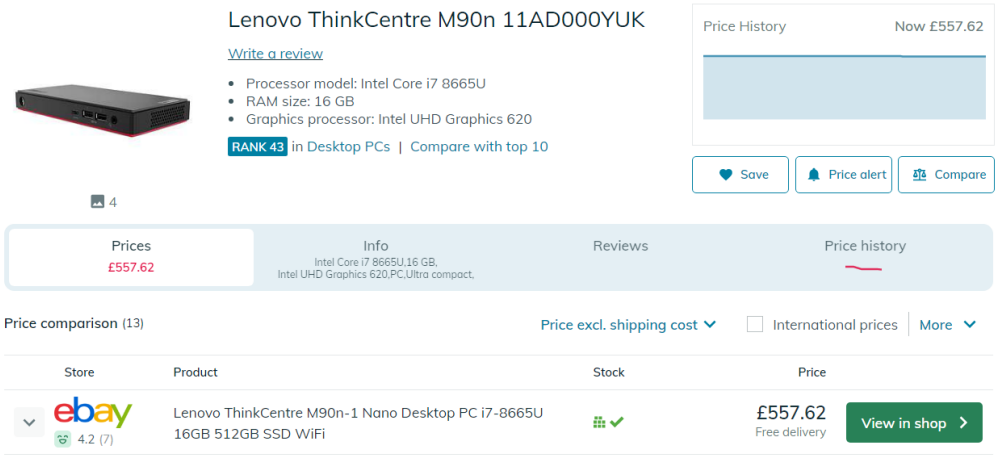
Gain performance and density in a compact 1U, 2-socket rack server for HPC, web tech and scale-out infrastructure.

Starting at £2,246.99
Ex. VAT @20%, free Delivery
Delivery information

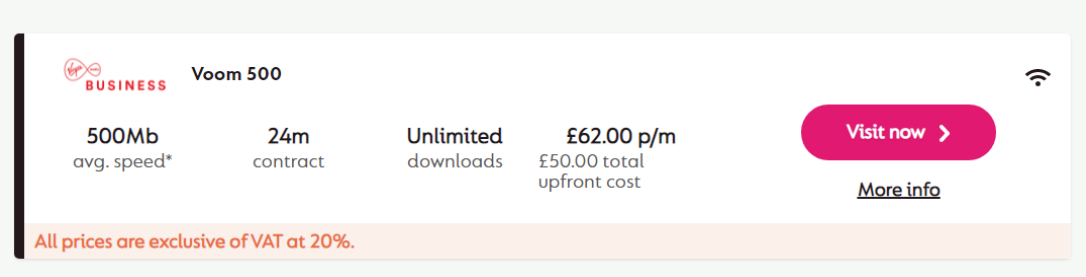
Instant Saving!
Save up to 35%

Add to Basket

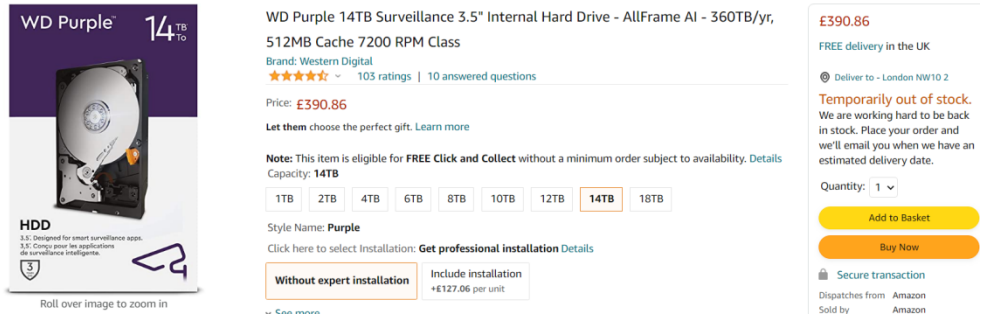
Desktop: 16GB RAM and i7 CPU. Price – 550 pounds.



Internet Virgin 500MB for business. Price – 62 pounds per month.



Hard drive – 14TB. Price 400 pounds.



In total system requires about 40K pounds for hardware. I expect that number can be increased for better servers or additional network line but not significantly.

4 Xeon servers	8800
50 Desktops	27500
Network 500MB (per year)	744
14TB 4 Nodes	1600
Total	38644

Conclusion

System described in this document might significantly improve health care services. Such system will provide much quicker help and might save human lives. All technologies mentioned here are widely used and I don't see any technical barrier that makes such system impossible to implement. Proposed technical design is known and popular in the enterprise world.

After all calculation system will require hiring 4 developers and will take about 1 year of development and about 200k pounds. Hardware cost estimated to be 40k pounds. All estimates are pretty optimistic but I expect that in the worst case it won't take more than 500k pounds.

I hope my work will be useful and improve the NHS IT services. If you have any questions about anything in this document you can contact me anytime.