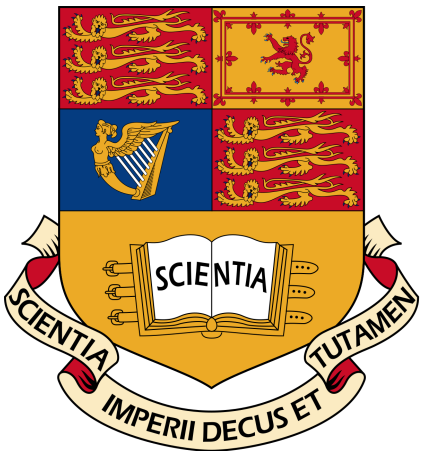

Imperial College London

Department of Electrical and Electronic Engineering

Giving Elderly and Minorities Access to Reliable Connectivity in the London Borough of Greenwich

Authors' Names:	***Authors' CIDs:***
Arnau Bonet Farres	01577271
Chidera Orabueze	01507473
Nima Karshenas	01500753
Seth Omoke-Enyi	01507158
Yun Hong Yong	01482411

Group 05 Supervised by *Dr. Javier Barria*

ELEC 96032 - Yr3 MEng. Electrical and Electronics Engineering Group Project

Final Report

DATE: 25th June 2021

Contents

1 Introduction

Connectivity is the ability of a computer, program, device, or system to connect to the internet. In the 21st century, we mostly use two types of connections: cable connection, and wireless connection. Cable connection can come from traditional coaxial cables or from the more modern fibre optic, and wireless connection is found in technologies like 4G, satellite connection, and one of the latest technologies: 5G. Even if connectivity has become a given for most of the people in developed countries, there are still contrasts in the presence and quality of an internet connection among regions.

Having access to a good internet connection has become necessary, because it gives access to information, opportunities and growth. According to a study by the Swedish government, the number of companies created increased by 1 for every 12,000 inhabitants where there was good connectivity versus where there was not[1]. We could say that connectivity has become a utility, just like water, electricity or gas, but some zones in big cities still lag behind in the development of a reliable and good quality infrastructure.

This issue was seen in the Canadian city of Toronto, where during the COVID pandemic lockdown, people living in low income areas without a reliable connection did not receive the important information and updates by the government in their personal devices. An initiative led by telecom company Cisco and the city council deployed a mesh network (see part 6) in 26 different buildings giving residents access to good quality connection. In the city of London, the borough of Greenwich is living a similar situation, the fibre deployment in the region is well below the averages of Europe as seen in the graph below.

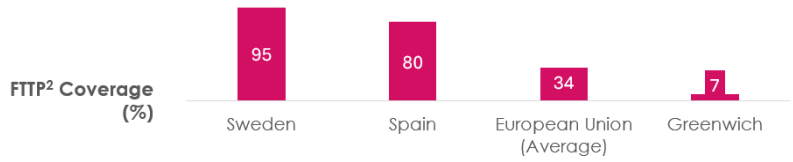

Figure 1: Comparison of FTTP coverage between regions

Also, the speed is well below the rest of the city, and is often too slow for basic utilisation of internet services.

Figure 2: Comparison of download speeds between London boroughs

As we pointed out previously, all the problems regarding connectivity were aggravated by the COVID pandemic, and the contrasts between regions were brought to light. Efforts are being made to deploy fibre to the property (FTTP) in Greenwich, to solve the “last mile problem”. This is a

phenomenon where the quality of the connectivity significantly drops in final stretch before reaching the homes, due to the technologies being used to connect the cabinets to the home. FTTP won't be fully deployed for some years at least, so we have been tasked with finding a good short term solution that could turn into a long term fix for connectivity in Greenwich. Of course, we will have to take into account not only technical feasibility, but also financial feasibility, because we will be working mainly towards increasing the connectivity in low income areas, focusing on larger, denser council estate buildings.

In this report, we will start by explaining the principles of 5G, why it will be the backbone of our solution, and why it is built for the future. We will then proceed to explain in detail a road map for the implementation of our solution, detailing the challenges and responses for every step. Finally, we will show the app we have built in parallel to our feasibility study. This app has the aim of interconnecting people in the borough and democratizing the access to information.

2 5G and its transition from 4G

2.1 Existing 4G Fixed Wireless Access (FWA)

As mentioned in the introduction, solving the last mile problem is what we have been tasked with. Fiber to the Cabinet (FTTC) is currently the most common form of connection used in the UK[2], meaning that copper cables are used for the last mile connection. The civil works required to change this connection to FTTP is the biggest logistical and economical challenge that is faced, meaning that extensive FTTP coverage may take years to be established.

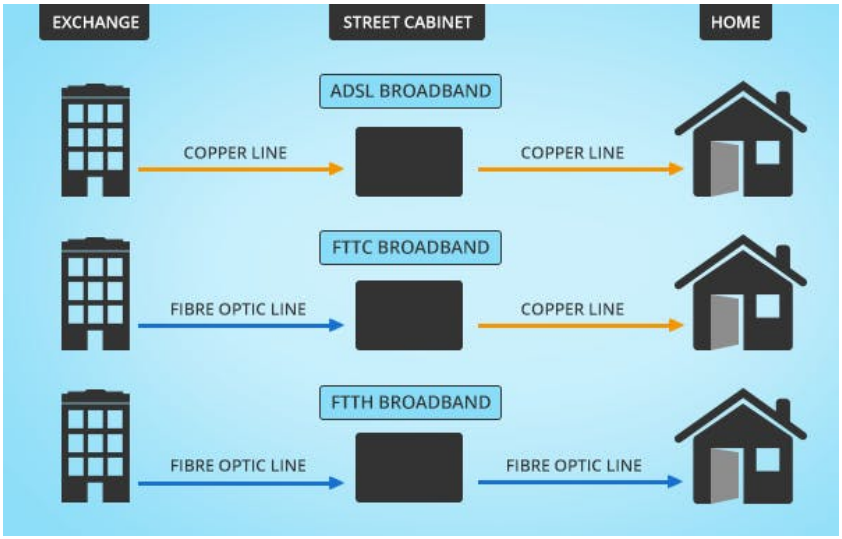

Figure 3: Diagram demonstrating the difference between ADSL, FTTC, and FTTP (or FTTH)

So what if wires are such a problem for the last-mile, why don't we just use wireless communication systems? Well, it turns out that this is actually already being used world wide, in the form of 4G FWA[3]. Since its deployment, the infrastructure and thus speeds of 4G have been constantly improving, in many cases, now surpassing the speeds achieved by FTTC [3].

Figure 4: The boom in Wireless Broadband(WBB)/FWA

2.2 Introduction to 5G

Now Where 4G's goal was to connect people, 5G aims to support a plethora of use cases, all of which fall under 3 umbrellas, and are summarised in Figure 5.

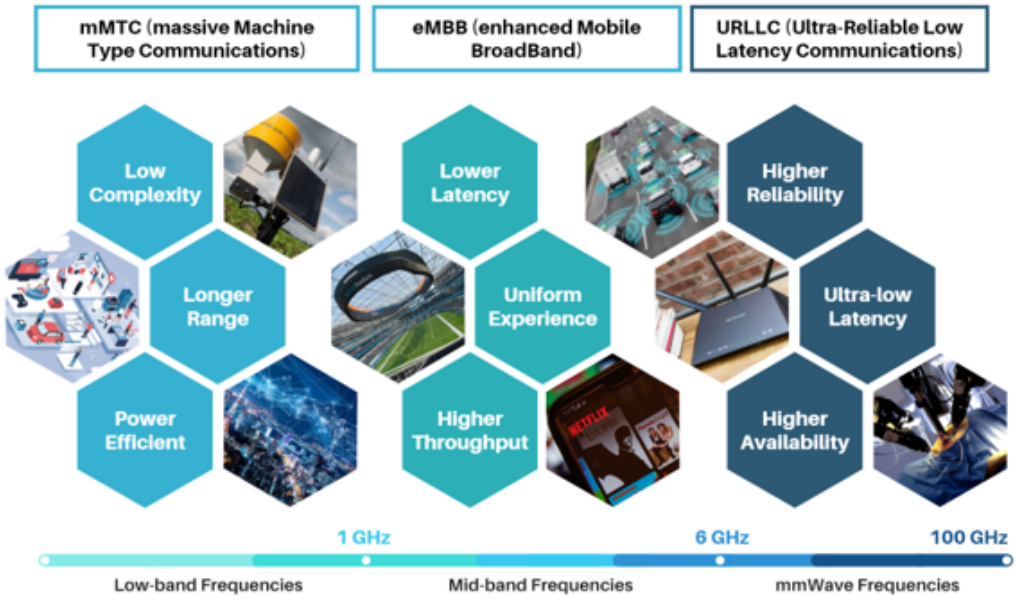

Figure 5: Overview of 5G spectrum

Unsurprisingly, the throughput of the network must massively increase to support the huge growth of connected devices to the network. To tackle this, designers of 5G have allocated different frequency bands depending on use requirements. The first of these use case umbrellas is massive machine type communication (mMTC), here devices are focused on low-cost, power-efficient, and long range communication. Lower frequencies have far better penetrative properties, meaning that they are relatively unaffected by obstacles such as concrete walls. This allows a single base-station to achieve far greater range compared to higher frequencies, reducing costs. Of course, at lower Frequencies, there is less bandwidth available, meaning that high data rates cannot be achieved, however, this is actually in the favour of power efficient devices, lower data rates means less sending and receiving data!

We have focused our efforts on the mid and high-band frequency spectrum as these have the capacity to facilitate the required throughput of home broadband use. The mid-band uses frequencies very similar to existing 4G LTE and so achieves similar coverage, but improved design means it achieves much higher speeds than 4G (100-900mbps)[4], this band is the current focus of Mobile Network Providers in the UK, as engineers can simply upgrade existing 4G sites, maintaining the already extensive 4G coverage.

The mmWave or high-band spectrum can support extremely high speeds (1-3 gbps)[4] but since its at such high frequencies, it suffers from 'blocking' which means that it requires line-of-sight communication, thus having a much smaller range per base station, requiring a denser network, for this reason its rollout will be much longer. In fact, the UK has only begun deploying low and mid-band systems[5], but plans have been made to begin auctioning mmWave bands[6]. This means that our immediate plans must only have the mid-band in mind, however, for a scalable solution, we must plan to eventually upgrade to the mmWave network to capitalise on the far greater speeds.

2.2.1 How higher throughput is achieved in the mid-band

As a basis of this section, we observe the throughput equation (1), to see what improvements were made to the network to support higher throughput in the mid-band despite similar available bandwidth to 4G LTE.

$$\overbrace{\text{Throughput}}^{\text{bits/s/km}^2} = \overbrace{\text{Cell Density}}^{\text{Cell/km}^2} \cdot \overbrace{\text{Available Spectrum}}^{\text{Hz}} \cdot \overbrace{\text{Spectral Efficiency}}^{\text{bit/s/Hz/Cell}} \tag{1}$$

To improve clearly either Cell Density, Available Spectrum or Spectral Efficiency must improve (or all). Increasing cell density can have huge financial costs, this would require digging up streets and densifying the fibre network in order to support the increase in cell towers, as for available spectrum, due to legal reasons and the fact that much of the mid-band frequencies are already in use, increasing the available spectrum is also too difficult to implement. This leaves the need to improve Spectral Efficiency, but how is this done?

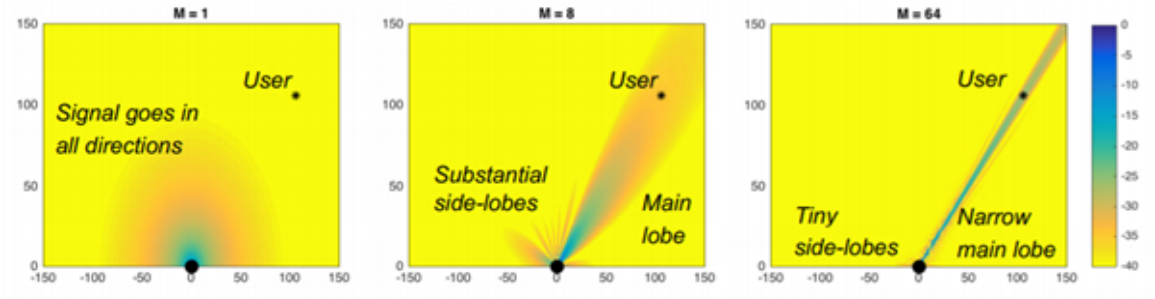

Figure 6: Multiple antennas for strong, directive signals. Note: M = the nuber of antennas in the array[7]

The main improvements made upon 4G LTE is the introduction of Massive MIMO (Multiple-input Multiple output) and 3D beamforming. As seen in Figure 6, multiple antennas can be used to increase

the strength and directiveness of the wireless signal. By adjusting the phase of the signal (using digital circuits) being fed to each antenna, engineers can have full control of the direction that the outward signal is being radiated. Where M=1, we can see that the signal is being radiated radially, in this case, the power dissipates $\propto \frac{1}{r^2}$, where M=64, the directional power is 64 times that where M=1, thus the signal received by the user is of far greater strength. This means that the channel carrying the signal is far more reliable (less variance), allowing the use of higher order modulation schemes, which essentially means that each signal carries more bits of information and thus a far greater spectral efficiency, as exemplified in Figure 7.

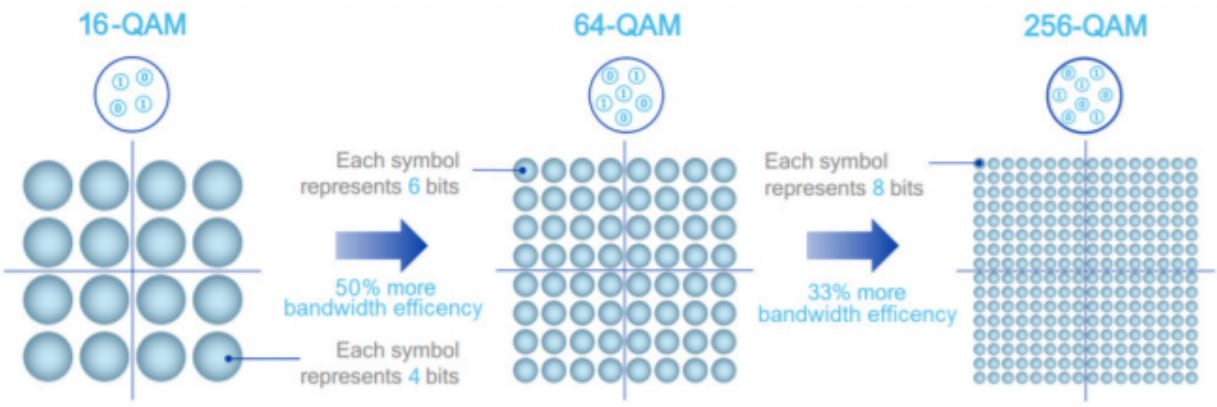

Figure 7: Higher order modulation schemes [8]

By having a more directive signal, this means that if parallel signals are being radiated, there is less chance of interference. Multiple antenna arrays also give us the freedom of increasing the orthogonality between channels, allowing the base station to fully separate the channels of different users, which is particularly useful in the uplink (Figure 8). In fact, if for example, M=200, 40 simultaneous single-antenna users can be supported[7].

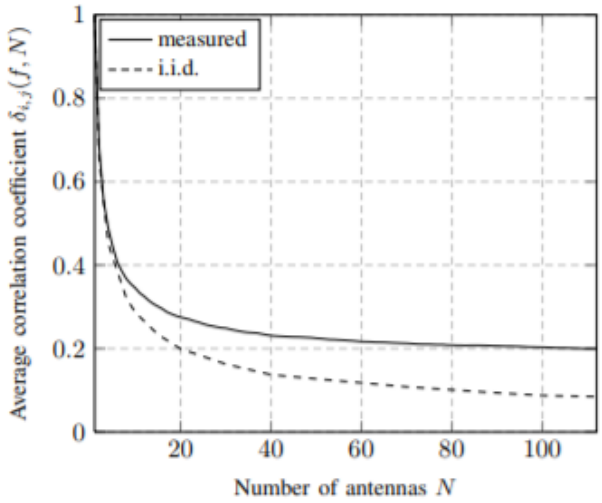

Figure 8: Channel Orthogonalisation[9]

This idea has been extended to the 3D plane, 3D planar antenna arrays means that full directiveness in 3D space can be achieved by an antenna (Figure 9).

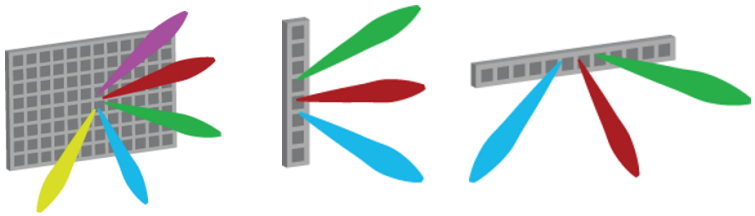

Figure 9: Diagram showing 3D and 2D beam forming. Note: Different colour beams correspond to parallel signals being radiated to different users [10]

So how does this play into our solution? Well, most of the improvements made to the network in order to support the upgrade from 4G LTE to 5G mid-band have been in the base-station (cell) domain, meaning no immediate work to the infrastructure is required to implement the upgrade. Installation of 5G cells can happen at existing 4G base-stations and can happen almost immediately.

3 Potential of 5G

3.1 Monetising 5G

The main problem ISP and telecom companies had with 4G was the inability to generate a lot of profit from it. Cash flows were satisfactory but the margin was very small. Competition between suppliers was very intense, and as soon as they started getting the benefits of economies of scale, customers demanded more speed and more data at the same or lower price.

Clearly the way forward in 5G monetisation will have to be different from what 4G was, ISPs will have to bring something new into the table. A report by strategy consulting firm PwC[11] highlights that the main potential is found in partnerships and bundles. Suppliers of 5G plans will make the most out of their infrastructure and service if they sell it together with other services. For example, a 5G service together with a subscription to a streaming platform. These partnerships can come in very different forms but the central issue is: who should be the company selling the bundle, the connectivity provider or the other service provider? That will depend on the case: in cases where the other service provider has a very deep market penetration then it will be more beneficial that they manage the customer relationships, but in cases where the the main service sold is the internet connection, then the ISP should manage the customer relationships. More detailed explanations can be found in the aforementioned report, but the main idea that we wanted to show is that 5G potential is very big, and the coverage in the borough of Greenwich will increase just by market dynamics.

3.2 Risk Assessment

The main risks associated with 5G are:

- Cybersecurity
- Supply chain complexity
- Issues related to public trust

3.2.1 Cybersecurity

Cybersecurity readiness is one of the fundamental issues holding up the advancement of 5G. As 5G relies on billions of interconnected devices, there is an increasingly heavy compute and network infrastructure that is needed to support 5G applications and services. The large number of devices connected to 5G networks increases the number of opportunities for attackers to sabotage a system or exfiltrate data using the weakness of a single device.

3.2.2 Supply chain complexity

Supply chains for 5G network components are long and complex[12]. Subcontractors are likely to be located in multiple countries, making it almost impossible to determine the national origin of a component. Hence, there are opportunities for malicious actors to infiltrate the supply chain at multiple points, either through a compromised vendor or by abusing insider access. Besides, patches or updates to the network also derive from a complex supply chain. Each software update or patch provides an opportunity to insert backdoors or accidentally introduce vulnerabilities into the source code. With current technology, it is impossible to exhaustively test every patch owing to time constraints, and actually proving that there are no defects is nearly impossible.

3.2.3 Issues related to public trust

The lack of public trust in 5G technology also serves as a risk that needs to be mitigated. More than 300,000 people and organisations from 220 nations had signed the 'Stop 5G on Earth and in Space' appeal[13], with some conspiracy theorists claiming that 5G is connected to the spread of the coronavirus. On top of that, anti-5G attacks on phone masts and general national security have caused 5G rollout to be delayed in most markets. Although no proven 5G health risks have been identified as of yet, numerous health concerns related to radio frequency exposure have surfaced. Therefore, Mobile Network Operators need to make sure the basic facts about 5G health risks are made widely available and presented in an understandable, non-technical way in order to mitigate risks associated with public trust.

3.3 Social Challenges in 5G Implementation

We have identified four main factors that might hinder 5G adoption, particularly in rural and low-income households. They include:

- Low awareness in the benefits of 5G
- Lack of clarity on 5G offering
- Lack of trust in 5G technology
- Structural limitations

3.3.1 Low awareness in the benefits of 5G

As the technology is still evolving and its value potential split across its different uses in different domains, there are difficulties in justifying the potential benefits of 5G to residents. These challenges in explaining the benefits of 5G home internet are further amplified due to the limits on in-person interactions in the COVID-19 era.

3.3.2 Lack of clarity on 5G offering

Different types of internet offerings (e.g. Fibre, 4G, DSL) are often promoted to some of the same households simultaneously and with similar branding, which could lead to confusion amongst residents. Some residents also find the process of installing an internet service to be challenging, often because of a discomfort with technology or the lack of technical support.

3.3.3 Lack of trust in 5G technology

As mentioned in Section 3.2.3 above, scepticism surrounding the cybersecurity and health aspects associated with 5G could hinder the rate of adoption. Additionally, concerns about future costs (e.g. hidden or unexpected fees for installation, equipment, and data usage) could cause residents to steer away from using 5G.

3.3.4 Structural limitations

Research shows that low-income households have experienced higher-than-usual levels of housing insecurity during the pandemic. Transience deters people from investing time in setting up internet service.

3.4 Best Practices to Encourage User Adoption

A study conducted by the Boston Consulting Group in 2021 identifies solutions and best practices to accelerate 5G internet adoption particularly in rural and low-income households[14]. These include:

- Strengthening communications
- Adapt the plan's design
- Expand support

3.4.1 Strengthening communications

To accelerate 5G adoption, any marketing and communication (from either the council or the internet service provider) should explain the 5G plan in clear, non-technical language. Flyers and marketing materials should highlight key eligibility messages, emphasize the service quality, and specify where to sign up. Phone calls, home visits, and flyer distribution in frequently visited locations (e.g. food pantries) are also effective means to boost awareness among those who are hardest to reach.

3.4.2 Adapt the plan's design

To further improve outcomes, installation instructions should be made clear with step-by-step illustrations of the installation process that are easy to follow especially for adults with limited technical experience. To boost installation success rates, certain community centres (e.g. schools, churches) could be selected as distribution points for 5G installation kits and provide technical resources to help with the installation.

3.4.3 Expand support

One-to-one guided assistance should be offered to eliminate confusion and overcome challenges related to a low level of technical literacy. Internet Service Providers can also help by having call centre teams catering specifically to 5G installation, to ensure fast, reliable, and effective support with minimal hold times. In the long term, it would also be helpful to help residents build their

digital literacy so that they understand how to use their devices, the internet, and various security and privacy safeguards.

4 Road map for implementation: connectivity in Greenwich

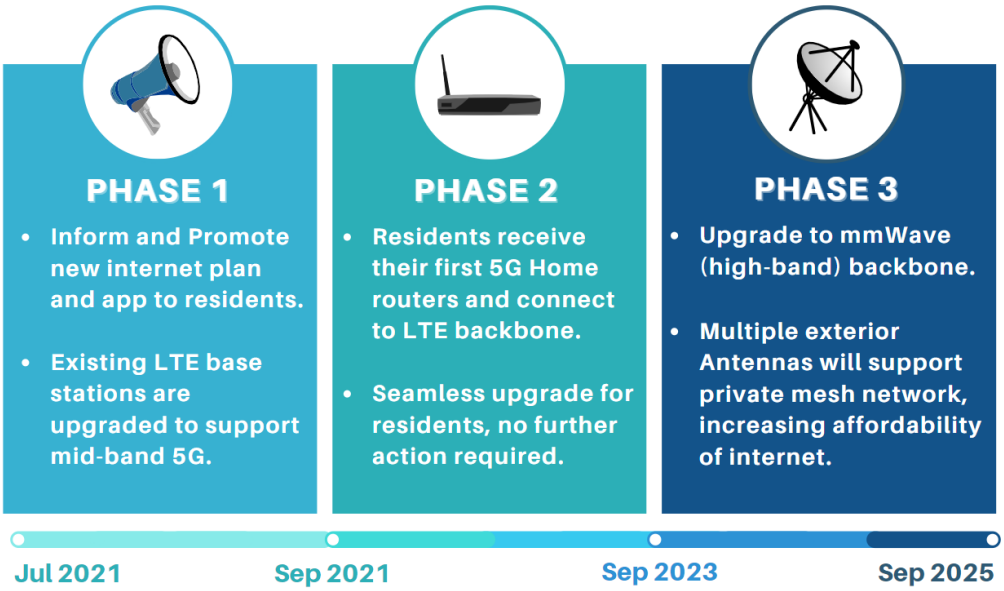

Figure 10: Overview of our 3 phase plan

4.1 Overview of the 3-phase plan

Based on the current growth of 5G coverage, we devised a three phase plan to ensure a seamless, and smooth transition to faster and more affordable internet in Greenwich. The phases are as follows:

- **Phase 1: Marketing** - Involves promoting 5G FWA offerings to Greenwich residents
- **Phase 2: Distribution** - Distributing 5G Home routers for self-installation thereby connecting residents to LTE backbone
- **Phase 3: Transition** - Installation of private wireless mesh networks in housing estates to further increase affordability.

4.1.1 Phase 1: Marketing

In this phase, we aim to increase awareness amongst the public and to convince Greenwich residents to adopt the suggested 5G FWA internet plan. Our main target will be low income households and sparser sub-urban areas that do not have access to fibre. The marketing strategy may include passing out informative fliers to residents. The flier would aim to address the misinformation surrounding 5G as well as improve public awareness as to the effectiveness of 5G.

4.1.2 Phase 2: Distribution

In this phase, residents would have access to the latest 5G home router plans. The mobile provider "3" offers a plan at £24/month with unlimited data usage, achieving speeds faster than existing Fibre to the cabinet (FTTC) providers. The Huawei 5G routers included in the package are plug and play

which means that users don't need to wait for the line to be changed or call for an engineer in the case or installation or repairs. This makes the process completely hassle free and would allow deployment to begin immediately. Three's 5G coverage has already reached parts of Greenwich, and the council can use this incentive on customers in order to grow the coverage into critical, low-income areas. This ease of setup ensure residents of all ages and technical abilities would be able get connected.

4.1.3 Phase 3: Transition

In this final phase of the transition, private wireless mesh networks would be installed in various council estates and buildings in Greenwich. To prevent the need for last-mile fibre installation, the mesh networks will be supported by multiple exterior 5G mmWave antennas, ensuring high-speed and low latency. Through the sharing of building resources, mesh networks provide a cost-effective and consistent distribution of internet across estates. Further details on mesh networks are provided in Phase 3: Mesh network with WAP

5 Phase 2: Huawei 5G CPE pro router

5.1 Technical feasibility

We intend to rollout phase 2 within the next 2 months, as seen in Figure 10. To ensure its success, there must be 5G coverage in critical, low-income areas by that time. We begin by observing current state of Three's 5G coverage in the Greenwich area. Pinpointed on the map is the estates in the Nightingale vale area in Woolwich, an example of one of our targetted areas (Figure 11).

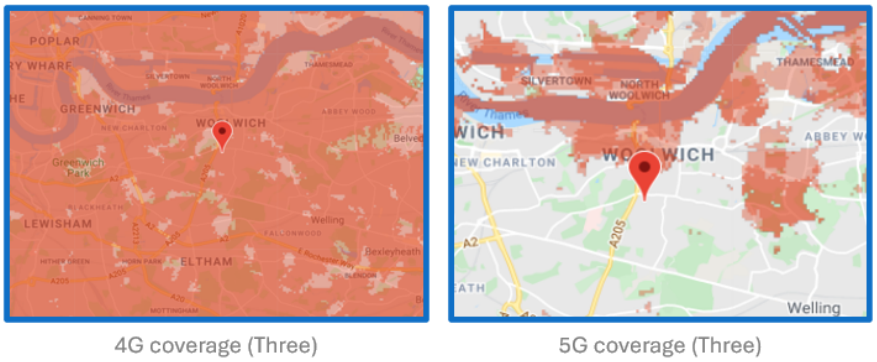

Figure 11: Three's 4G and 5G coverage in Greenwich [15]

It can be seen that there is already full 4G coverage in the Greenwich area, with 5G coverage already sprawling into large parts of the area as well. The Nightingale Vale estates appear to be in an area where coverage will naturally grow into, and as mentioned in Section 2.2.1, upgrading existing 4G sites to 5G can be done immediately, subject to the network providers wishes. If the council approaches Three with the intention of providing a substantial customer base, this incentive can entice Three to accelerate their expansion of the 5G network in Greenwich.

As mentioned before, the recommended plan is a 24 month contract priced at £24 per month, with unlimited data usage and a promised average speed of 100 Mbps, faster than FTTC options as highlighted in Figure 21. This baseline speed will only increase with time, especially since Three actually owns the most mid-band bandwidth than any other provider[16], and have plans to further

improve the fiber backhaul supporting the 5G network[17].

The 5G home router included in this plan comes with many logistical advantages over a traditional home Wi-Fi router. First of all, as alluded to before, it's plug-and-play, this means that not only do residents not need to call an engineer, but they can take the router wherever they want, as long as there is a plug point! This means that if stronger internet is required in a bedroom, they can simply take the router to that room for potentially faster internet.

5.2 Financial feasibility

Figure 21, shows Three's 5G home router contract compared to existing tradition home broadband plans in the Nightingale Vale area. It is already cheaper than the other options, and promises faster average speeds. Since 5G FWA is only in its infancy, not only will we see the prices fall, but the speeds will undoubtedly increase as well, it seems like an unquestionably better option for residents.

To reiterate, the expansion of 5G requires no significant extra civil works, this means that the council can can avoid the mass disruption that digging up streets can cause, as well as the huge costs that go along with it.

5.3 Security concerns

Huawei has been involved in some major headlines in the past years. Former United States of America president Donald Trump prohibited the use of Huawei technology in any of the country's public infrastructure and publicly accused the company of espionage. Despite this, none of these accusations have been backed by any type of proof. The hardware has been shown to be as faulty as any other provider. The truth is, Huawei has found itself stuck in the middle of the trade war between China and the US, so it has been used as a punching bag by the Americans even if their accusations were not based on any proof.

Two main accusations stood out: Huawei technology had secret back doors so that the Chinese government would be able to control it if there was a war between the two countries, and also that this technology had some secret code that would completely shut it down, so that China could start a communications war with the US. As we pointed out previously, these accusations have not been backed by any proof, and if the main issue is that the technology is fabricated in China, then there is not one major electronics provider that is safe from it. Huawei is as safe as any other provider.

6 Phase 3: Mesh network with WAP

6.1 Introduction to Mesh Networks

A mesh network is a local network topology in which each node in the network connects directly and non-hierarchically to the other nodes and communicates efficiently with each node to route data to and from the clients connected to the network.

Software is used to program the nodes and instruct them on how to communicate with the larger network. Information travels across the network from point A to point B through the quickest and safest path in a process known as dynamic routing. Dynamic routing is the process whereby the mesh network creates multiple routes between its nodes. This allows the network to function effectively if one network node fails; data moving across a network will have another path it can use. This also applies to wireless networks as a wireless signal can take multiple routes to reach its destination.

Mesh networks were developed to alleviate the constraints that arose from the use of repeaters, extenders and boosters

Types of Mesh Network

- **Full Mesh Network topology:** In this type of network, every node in the network connects directly to other nodes in that network. The number of nodes can be calculated using the formula n(n-1)/2. This connection yields the greatest amount of redundancy.
- **Partial Mesh Network topology:** In this topology, at least two of the nodes have connections to multiple other nodes that are in the network. It is cheaper to implement but has less redundancy[18].

Advantages of Mesh networks

- **Independent:** The lack of dependency on any particular node allows every node to participate in the relay of information.
- **Self configuration:** In the event of a node failure, the nodes self configure to allow dynamic distribution of workloads. This ensures that the network stays online in the case of isolated node failure. It also helps to reduce maintenance cost of the network.
- **Consistent network credentials:** In a mesh network, all the nodes share the same SSID and password as part of a single wireless network. This eliminates the need for users of the network to reconnect as they move across the building and switch from one router to the other.
- **Stronger network signals:** The nodes in mesh networks produce stronger WiFi signals that work through walls better as compared to repeaters and extenders[19].

Disadvantages of Mesh Networks

- **Price:** a full mesh network can be expensive to set up due to the larger number of components needed.
- **Redundancy:** There is a risk of redundant node connections in the network setup. Two or more nodes may be providing WiFi to the same area and as such increasing the cost of the initial network setup[20].

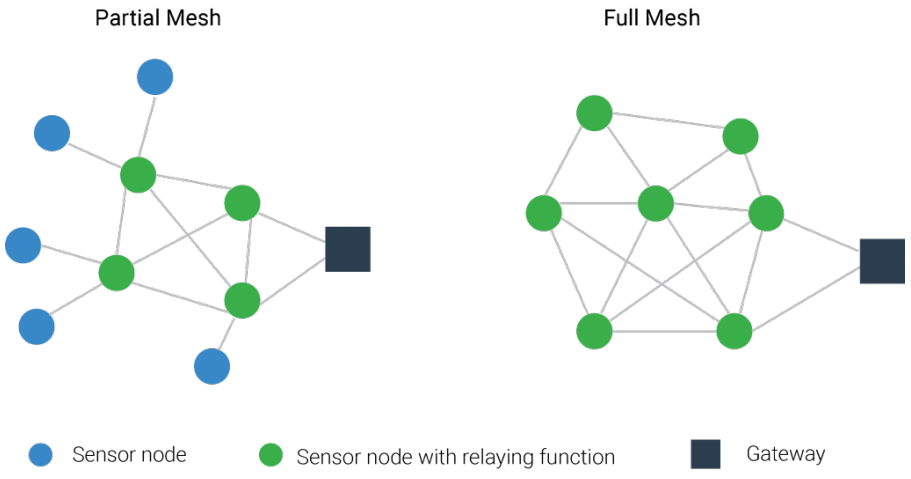

Figure 12: Diagram of a Full and Partial Mesh Network

6.2 Star vs. Mesh Topology

Star Topology like the name implies is a network that is configured in the shape of a star with the central core and other nodes connected directly to that core.In a star topology, the nodes do no communicate with each other. Instead each node passes messages to the central core and the central core then passes it on to the recipient(s). This topology works well for small networks but as the size of the network increases, there is a risk of server overload and slower transmission of data.

Advantages of Star network

- A fault on one node doesn't affect the general performance of the network.
- There is no disruption to the network when devices are being connected and disconnected.
- The centralised nature of the network makes it easy to manage.

Disadvantages of Star network

- If the main core suffers a fault, the whole network is disrupted and cannot participate in the network communication.
- Data transmission can be a bit slow because there is only one route for data to flow back and forth between devices. The data can only be transmitted via the central node.

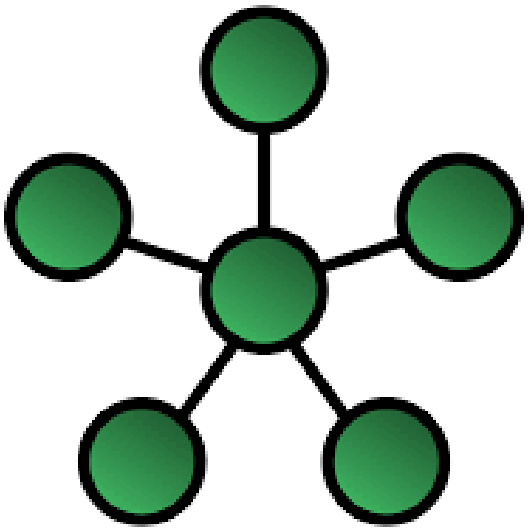

Figure 13: Image of Star Network topology

6.2.1 Choosing between the mesh and star network for the council homes

Considering the nature of the mesh and star network, together with the layout of the council homes in Greenwich, mesh network is the better topology to use in the council home. The mesh network would have nodes located at strategic places in the home. Due to the dynamic nature of the mesh network, the various nodes would transmit WiFi signals between each other using the most efficient route. This is unlike the star network which only uses one route to transmit data (Wi-Fi). There is also very little risk of overloading the mesh network because there is no central node needed to handle

transmission of data.

The nodes we would employ in building the mesh network is the Orbi RBK853 Tri-Band Mesh WiFi System which can handle up to 100 devices at a time. This capacity is more than suitable for the residents of the council home. Furthermore, there is currently no wireless star network devices available on the market. Given that the main aim of the project is to reduce cost for the council as much as possible, a wireless network would save on costs required to drill and dig holes to pass the wires through the building. As such a mesh network is all the more preferred to use in the council homes. In the next section, we would give a more detailed plan on how the mesh nodes are placed around the house.

6.3 Implementation: an architecture example

To better understand how Mesh Networks can be used to provide high speed access in buildings, we are going to consider an example of a large building with several flats, similar to other large/council homes in the Greenwich Area. The building under consideration is known as Munkenbeck Building and it is located in Paddington. It was built in 2004 by Stephen Marshall Architects, to provide affordable housing to residents in the Westminster region. It has 9 floors and houses 80 apartments.

Figure 14: Munkenbeck Building

Our chosen Mesh Wi-Fi system consists of a main router node, supported by an external antenna, that connects directly to the modem/router (the Huawei 5G Router in our case), and a series of satellite modules, or nodes, connected to AC power and placed around the building to extend full Wi-Fi coverage. We will estimate the number of Mesh Wi-Fi nodes required to provide connectivity across the whole building without any dead spots, and the cost estimates per household for this setup.

For our calculations, we rely on one major assumption that every British household has access to 10 connected devices[21]. So for the floorplan shown below which contains 8 apartments, the Mesh WiFi system must be able to support a minimum of 80+ devices.

Below is a diagram of the floor plan for Floor 1 of the building. We will estimate the number of mesh Wi-Fi nodes required and the cost of infrastructure and setup for the floor and extrapolate the figures to provide an estimate for the entire building.

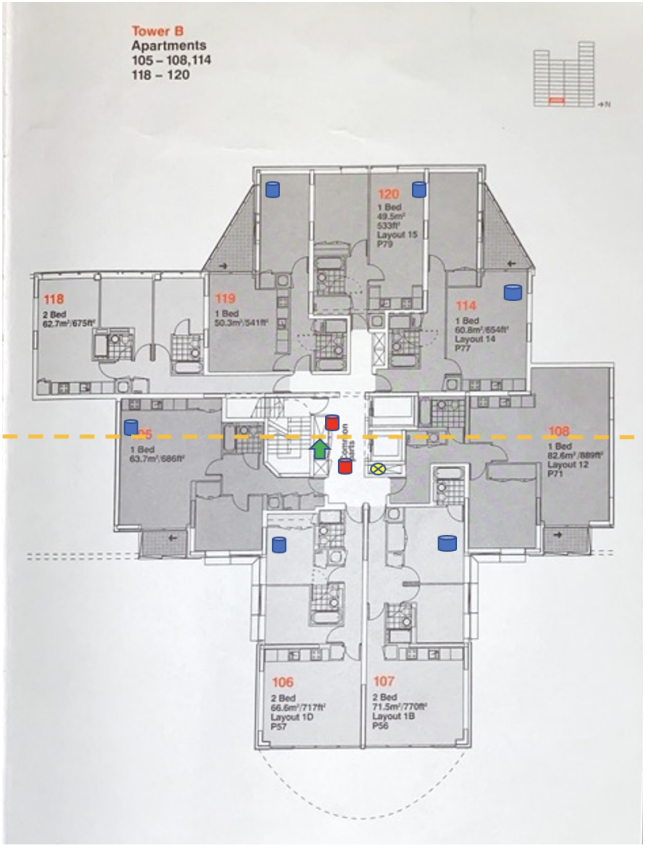

Figure 15: Floor 1 plan

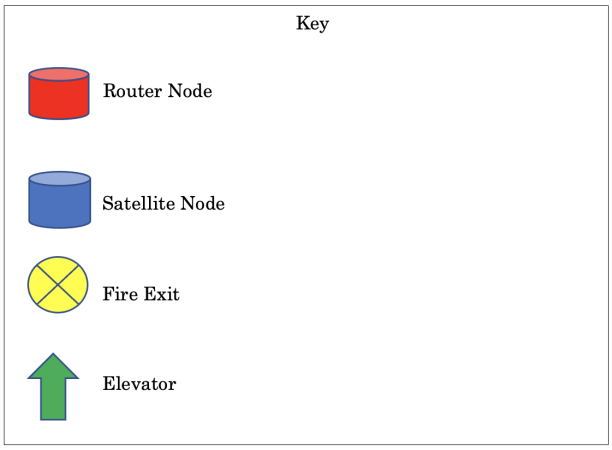

Figure 16: Key to floor plan

The main factors we had to consider in the strategic placement of Wi-Fi access nodes were:

- **Area of the floor**: Each Wi-Fi Mesh system has a maximum range of 6000 square feet, so we have to ensure that the floor plan is less than the maximum range or we utilize multiple systems per floor. The area of the floor above is greater than 6000 square feet so we opted for the latter.
- **Physical factors**: Physical obstructions like the walls between flats(especially if they're made out of concrete), fire exit doors and elevator doors can seriously obstruct our Wi-Fi signal. To

ensure these do not mitigate our connectivity, we used two Mesh Wi-Fi systems for the floor plan above, with four nodes serving each side of the floor demarcated by the dashed yellow line.

As can be seen above, for floor 1 we would require a minimum of 8 Mesh Wi-Fi nodes. Although a set including 1 main router and 2 satellite nodes is enough to cover the whole of floor 1 (5796 sq.ft.), it does not have the capacity to support 80 devices, hence the need for 6 further nodes. These extra nodes will of course add to the overall cost. However, by having an 'excess' of nodes, we ensure that a consistent and strong signal is experienced throughout the building. The main nodes are represented by red cylinders while the satellite nodes are represented by the blue cylinders. For the entire building, we estimate we would require 50 mesh Wi-Fi nodes(main and satellite nodes inclusive). A breakdown of the number of nodes required per floor can be found below:

Estimated Number of Wireless Access Points required	
1st floor	8
2nd floor	8
3rd floor	8
4th floor	8
5th floor	4
6th floor	4
7th floor	4
8th floor	3
9th floor	3
Total	50

Figure 17: Floor Breakdown

N.B. Floors 4-9 require less nodes because they have less apartments. As a result, less devices are required to provide connectivity for.

6.4 Financial feasibility

The approximate price for the "NETGEAR Orbi Whole Home Mesh Wi-Fi System" is £310 with £187 for each extra satellite node[22].

For floor 1, we have already established that we would require 8 mesh nodes i.e.(2 main nodes + 6 satellite nodes). Since each system comes with a pack of 3 nodes i.e. (1 main router and 2 satellite nodes) we would need to purchase two more satellite node add-ons The price breakdown of floor 1 can be found below:

Price Estimation for 1st Floor	
Price for 1 Wi-Fi Mesh System x2	£619.98
Add on wireless access point x2	£373.76
Total cost for 1st floor	£993.74

Figure 18: Floor 1 Price Breakdown

Using this figure, we can now calculate the price estimate for the entire building since we know the number of apartments for each floor and the number of nodes required per floor. The price estimate for the Mesh Wi-Fi infratstructure can be found below:

Price Estimate for entire building		
1st – 4th floor	£993.74 x 4 + £800 =	£3974.96
5th – 9th floor	£309.99 + £ 186.88 + £400= £496.87 x 5 =	£2484.35
Estimated Total		£ 7659.31

Figure 19: Building Price Breakdown

So it would cost a total of £7660 to obtain Mesh Wi-Fi Technology for the entire building, but other costs must be taken into account as well before we cam estimate the price estimate per household. These are divided into One-Time and Running costs.

The one time costs are outlined below:

- **Infrastructure Costs**: This is the total cost associated with obtaining all necessary Mesh Wi-Fi infrastructure. We have already calculated this to be £7660 for the entire building.
- **Set-Up Costs**: This is the total cost associated with setting up the Mesh Wi-Fi technology in each apartment. The approximate price for set-up is £67 per system.
- **Contingency Costs**: We have set up a contingency cost as a Risk Management technique to account for uncertainties as to the precise content of all items in the estimate, how work will be performed, what work conditions will be like when the project is executed and so on. The contingency cost is 5 percent of the total Infrastructure + Setup costs.

The running costs are as follows:

- **Internet Service Costs**: This is the recurring cost paid to the internet Service Provider(ISP) for access to the internet. Because internet resources are shared between different households, this figure amounts to £6 per household per month.

- **Maintenance Costs**: This is the total cost associated with maintaining the Mesh Wi-Fi infrastructure. Our estimated value for maintenance costs is £5 per household per month.

Adjusting for all these costs, each household will need to pay £16 per month for 5G internet in Phase 3, which is not only the best value for money on the market, but also the best speed on the market. This added £5 per month to the running costs has been chosen such that the council can repay the upfront costs over a period of 24 months. Once repaid, subscription can be reduced even further. This affordability matches the focus and target market of the project: lower-income areas with no access to FTTP.

Budget Item	Cost per household (£)
One-time Costs	
Infrastructure	£96
Setup	£15
Running Costs (per month)	
Internet Service	£6
Maintenance	£5
*Total Cost Per Month** **Assuming 24-month contract*	£16

Figure 20: Costs per household

Below is a comparison of our proposed plan which spans the entire phase 3 period to other plans offered by ISP's in the Greenwich region.

	Now Broadband	Sky	PlusNet	Three	Our Proposed Solution
Type of Service	Wired Broadband	Wired Broadband	Wired Broadband	5G (Phase 2)	**5G + Mesh Networks (Phase 3)**
Monthly Cost	£28	£28	£26	£24	**£16**
Average Speed	36 Mbps	59 Mbps	66 Mbps	100 Mbps	**500 Mbps**
Contract Length	12 Months	18 Months	18 Months	24 Months	**24 Months**

Figure 21: Comparison of proposed plan with other service providers

*Please note that all prices quoted in this report may not represent actual values rather they are a price estimate for the cost of setting up a mesh network in the aforementioned building.

7 Mobile Application to connect isolated residents

7.1 Design of App

7.1.1 Motivation behind the introduction of an App

The project title given to us by DG Cities reads as: "COVID-19 and better connectivity". From preliminary discussions held with our in-company supervisor; Balázs Csuvár, we came to the conclusion that our proposed solution to this brief did not necessarily have to be restricted by a plan to only bring physical internet connection to the residents of Greenwich. Equally important is the task of notifying and teaching residents on how to connect to the newly installed high-speed internet as well as providing a centralised platform for simpler, streamlined communication between the council and its residents, and between the residents themselves.

COVID-19 has dramatically increased the demand for such a platform, with many residents, especially the elderly, being isolated within their homes, restoring a sense of community, albeit virtually, is of utmost importance. Furthermore, the council has highlighted the need to distribute important COVID-19 information. A mobile application is ideal for implementing these features in a centralised, organised, and ergonomic way. In the following sections, the process of designing the app to adhere to the needs of the council will be explained.

7.1.2 Outline of product expectations

Prior to design, we felt it was important to clarify, define, and formalise the set of criteria expected by the client to be achieved with the mobile application, these have been summarised as follows:

- Ensure each feature is simple to use for for all ages and demographics.
- Implement a feature that displays important council-published information, and ensure the upload of articles/stories can be done quickly and without prerequisite technical knowledge.
- Provide a platform where residents can socialise, preferably through voice rather than messaging.
- Implement a portal where residents can voice their concerns and report dangers to the council.
- Ensure design is consistent, and avoids being overwhelming and cluttered.
- Cross-platform capabilities to ensure all residents have access to services.
- All features to be to be implemented on one app with one login allowing access to all features.

These set of criteria will act as the backbone behind the design and implementation of the app, each decision will refer back to this to ensure that it is inline with the vision of the client.

7.1.3 Design Specifications

With an outline for what is expected from the app set, we can now begin to conceptualise its design. For simplicity, the design specifications have been divided into two sets of agendas. The first, concerns the functionality of the app, this will encompass all the features, how the back-end of the app is managed, and how the features are integrated together. The second set consists of the aesthetics and styling of the app, what colours are used, what font sizes, how each page and feature looks and feels, etc., both being summarised below:

Functionalities:

- **Sign-up, log-in flow:** Simple login flow with username, email, and password with secure storage of this information, potentially the addition of two-factor authentication for added security.
- **Home Screen:** When the user has logged in, the app should open up on a page holding stories/articles published by the council. This will be the page where important COVID-19 information will be published. By opening up the app on this page, residents see the important information before moving onto the other functionalities of the app. Stories should be organised in a scroll-able list where the main information regarding each story is contained within a tile, when the tile is pressed, the user should be navigated to a page where they can learn more about the respective article.
- **Chat Room:** The Chat Room will contain a set of Administrator-created channels where residents can join any of these and engage in conversation with the users already inside the channel. Each channel should be displayed on the screen along with all the names of the users inside the channel. The displayed name of the users should be their full name or username that was used when signing up. This page should have simple user-control buttons for muting and silencing audio. The process of joining a channel should be kept as intuitive and simple as possible.
- **Contact Us Screen:** This screen will contain a simple messaging page between the user and some back-end handled administrator. The design will largely be based on messaging apps such as WhatsApp and Facebook Messenger. At the top of the page there should be an option to call the contact centre of the council. The Application Programming Interface (API) used as the back-end for messaging should be secure and ideally encrypted.

Aesthetics and Styling:

- **Colours and Themes:** The colours used in the app are based off of the London Borough of Greenwich's website, thus the main colour has the following hex code: "#c20027", this will be complemented with simple Blacks and Whites to prevent being overwhelming.
- **Font and Font Size:** The main font family that will be used is 'Helvetica', this is a simple, clear and modern font, perfect for this setting. Fonts will be bold and with a minimum size of 16px in order to be clear and easy-to-read for the elderly.
- **"Look and Feel":** For the most part, small touches to styling such as dividers, containers and shadows will take inspiration from existing social media applications in order to keep the feel of the application as modern as possible.

7.1.4 Design and Sketches: Home Screen

With reference to the specifications for the home screen defined in Section 7.1.3, the application *Procreate* was used to sketch up the design of the Home Page, shown in Figure 23. To prevent an overwhelming amount of information, the tile will contain the crucial information regarding the article, if the user is interested, then they can press the tile to find out more information.

The design takes some inspiration from the app, *Apple News*, this is due to its simple design and the fact that its design has a similar use case in mind, informing the every day user. Figure 22 shows a summary of the features that inspired and contributed to the design of the Home Page. Despite its sleek design, the page feels cluttered with tiles of various sizes and the page changing layout

constantly as the user scrolls. For our design we will opt to use a single column scroll layout to avoid cluttering the screen and making sure it's easy to follow.

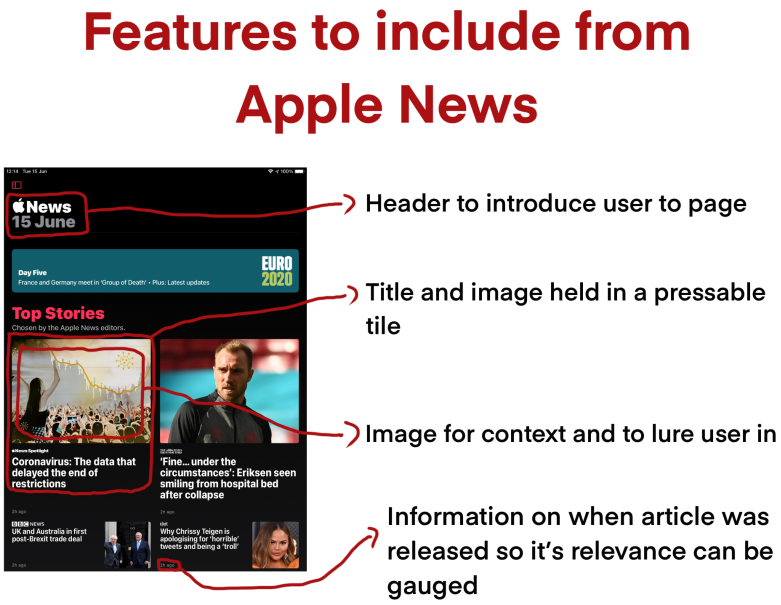

Figure 22: Feature analysis of Apple News

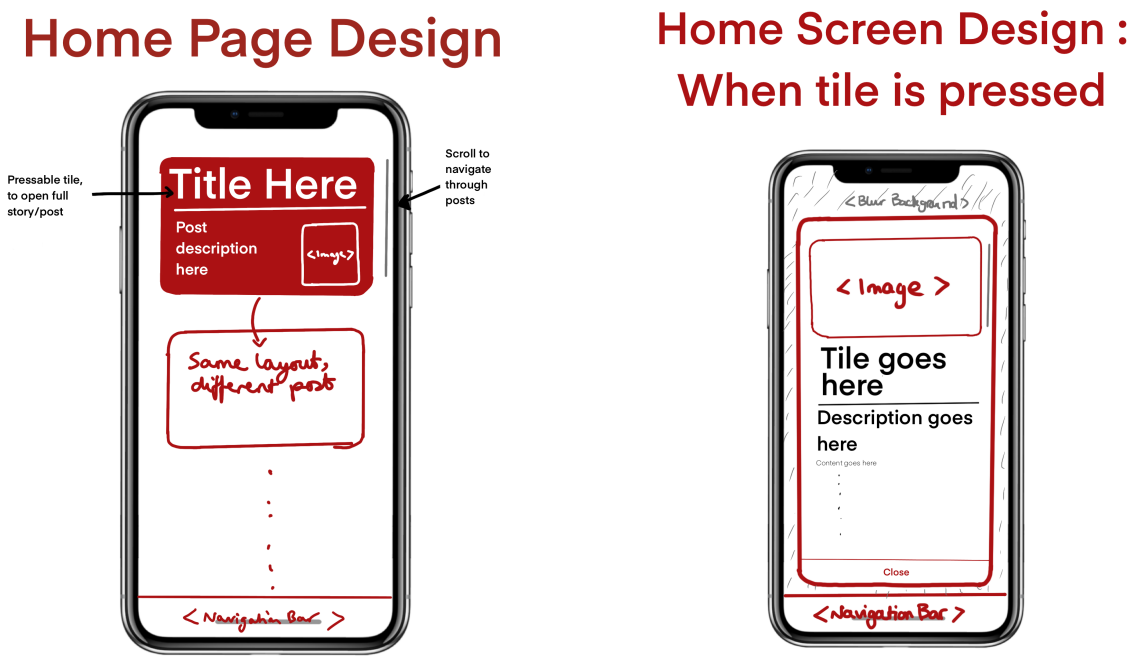

Figure 23: Sketches outlining the key components of the Home screen's design

As seen in Figure 23, the same colouring scheme outlined in section 7.1.3 is used, with a tile holding the image, title, and description of the article, and its size changing dynamically according to the content inside it. When the tile is pressed, a pop-up will contain the full contents of the story,

with a close button for an unassuming means to exit from the pop-up.

7.1.5 Design and Sketches: Chat Room

Much like with the Home Screen/Page, the design of the chat room follows the criteria set out in Section 7.1.3.

The design takes inspiration from the gaming voice and messaging app Discord. Many of us from the group enjoyed nights of talking to our friends during lockdown on this app, taking inspiration from its design, we believe it can help recreate such an atmosphere. Its most notable features to include are highlighted in Figure 24.

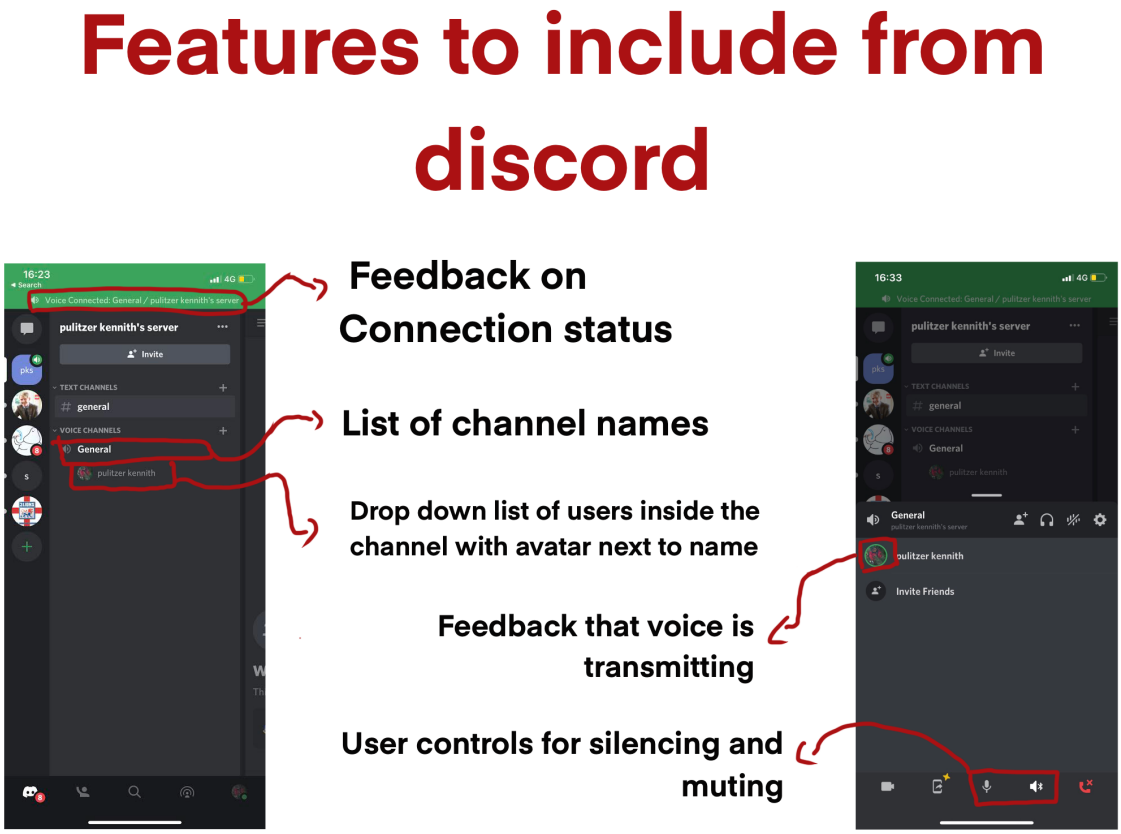

Figure 24: Feature analysis of Discord

It must be noted that the target audience of Discord is young gamers, for this reason its user interface has many features and controls, allowing gamers to fully customise their experience. We, on the other hand, want to design the interface such that no tutorial would be required in order to use the functionalities, for this reason, only the simplest features will be adopted in the design of the Chat Room. All the features listed in Figure 24 are included in the final design shown in Figure 25, but instead of requiring user navigation to access these features, they will all be available on one page. In terms of the process of joining a channel, in Discord this is done by pressing the channel name on the UI. This, we believe, will be confusing to many first-time users. To avoid this, we prompt the user to select the channel they would like to join from a dropdown list, and to join, press the 'Join' button. This feels like a more intuitive approach to the process.

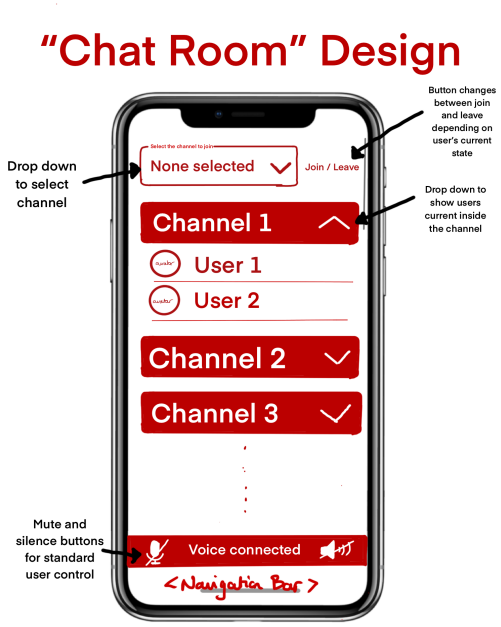

Figure 25: Sketches outlining the key components of the Chat Room's design

7.1.6 Design and sketches: Message us

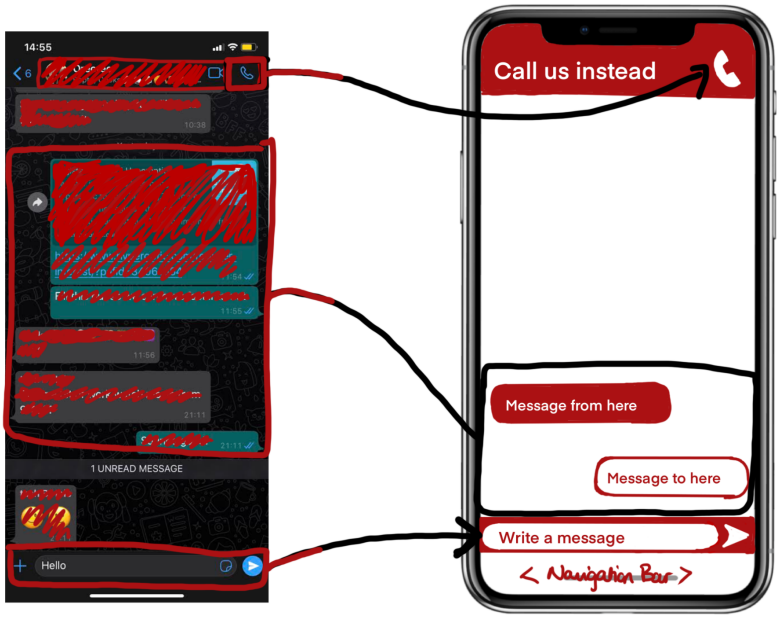

Figure 26: The design of the Message Us page compared to Whatsapp

The design of this page is heavily influenced by apps like WhatsApp and Facebook Messenger, these apps have become so heavily engrained in our way of life, that if we decide to move away from such a design then we risk confusing first time users. Instead, the layout will be the same but some

features such as video calling, attaching documents and images has been removed to streamline the interface. If users begin requesting more features, these can of course later be added in. Figure 26 shows the design of the Message Us page compared to WhatsApp.

7.2 Implementation of App

Note: The full code for the App can be found on our GitHub repository: `https://github.com/nimakarshenas/GreenwichTogether`.

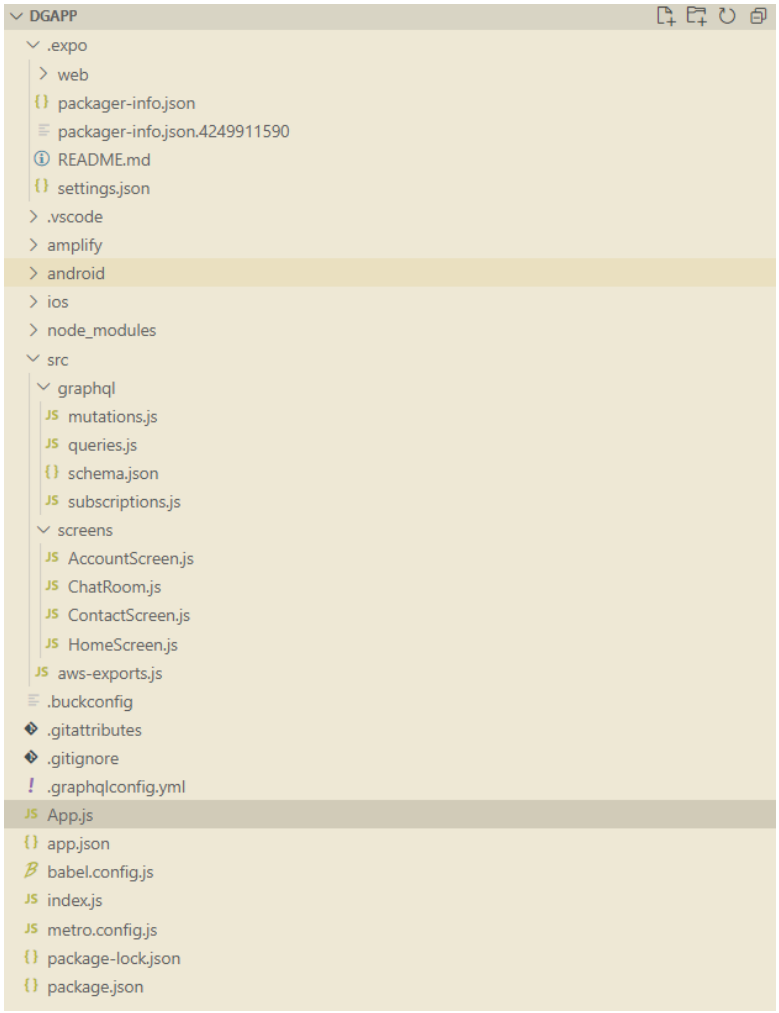

Figure 27: Folder organisation for reference throughout this section

7.2.1 Overview of implementation strategy

The Mobile Operating System Market Share is split almost 50-50 between Android and iOS[23], taking this into account, having a cross-platform app was essential so that no resident is left out from accessing features. Nima, being the chief developer already had some experience using the React Native JavaScript package for multi-platform app design and so this was decided to be used as the platform for the front-end of the app development.

In terms of the back-end, the following summarises the service/platform used for each section of the app:

- **Login Flow:** Amazon Amplify, a branch of Amazon Web Services (AWS) will act as the back-end here, this was chosen for the high level of security of AWS, ensuring that the information of all the users are as protected as possible. Amplify also has a default authentication UI design that handles all the authentication methods and communication between the app and the back-end. This will help save crucial development time for this section of the app so more focus can be put on the main features. Furthermore, without using the advanced security features, use of this service is free.
- **Home Page:** Amplify will again be used for the uploading of articles from the web to be displayed on the app. The first reason for selecting AWS is that it is already being used to handle authentication, the uploading of data through GraphQL; having the back-end under one umbrella can be easier to maintain for admins. The second reason is that updating of the data schema, uploading of articles, updating of code can all be managed within the Amazon Admin UI online, this will be far easier to maintain, especially with a small admin team.
- **Chat Room:** When looking for a service to host this feature, AWS did not have the required infrastructure to implement a 'voice channel', for this reason, Agora.io was used. This SDK has in-built functions to handle all the tasks that were chosen to be implemented in Section 7.1.5. It also offers very attractive pricing (see Section 7.3.4). AWS will be used to keep track of the names of users that are inside the voice channels, the creating of new channels, storing of channel tokens for Agora.io as well as the channel names.
- **Message us:** GraphQL is also suitable for this feature, just like the Home Page, thus for the same reasons, Amplify will be used again.

7.2.2 Login Flow

Since we are using the built-in Login flow provided by Amazon Amplify, the implementation of this section is trivial. We begin by installing all the required files and setting up the backend, this is done by following the Amazon Amplify Authentication documentation[24]. To include within the app, we wrap the main App component (in App.js) with the authenticator provider as follows:

```
export default withAuthenticator(App, {
  signUpConfig: {
    signUpFields: [
      { label: "First Name", key: "name", required: true, type: "string" },
      { label: "Family Name", key: "family_name", required: true, type: "string" },
    ]
  }
});
```

Through the `signUpConfig` attribute, we also add First Name and Family Name sign up entry fields so that they can be used to be displayed when a user joins a channel.

7.2.3 Navigation Bar

The navigation bar was setup at the bottom of the screen to avoid being obstructive, and to allow users to navigate users between the 3 screens in one press. The code to implement this was included in the main App component and is included below for reference if more screens are to be added. We

have included an optional screen that is commented out, the Account Screen, as a guide to adding extra screens to the navigation.

```
function MyTabs() {
  return (
    <Tab.Navigator
      initialRouteName="Home"
      tabBarOptions={{
        activeTintColor: '#c20027',
        inactiveTintColor: '#ff99ad'
      }}
    >
      <Tab.Screen
        name="Home"
        component={HomeScreen}
        options={{
          tabBarLabel: 'Home',
          tabBarIcon: ({ color, size }) => (
            <MaterialCommunityIcons name="home" color={color} size={size} />
          ),
        }}
      />
      <Tab.Screen
        name="Chat"
        component={ChatRoom}
        options={{
          tabBarLabel: 'Chat Room',
          tabBarIcon: ({ color, size }) => (
            <MaterialCommunityIcons name="account-group" color={color} size={size} />
          ),
        }}
      />
      <Tab.Screen
        name="Contact"
        component={ContactScreen}
        options={{
          tabBarLabel: 'Message us',
          tabBarIcon: ({ color, size }) => (
            <MaterialCommunityIcons name="message" color={color} size={size} />
          ),
        }}
      />
      {/*<Tab.Screen
        name="Account"
        component={AccountScreen}
        options={{
          tabBarLabel: 'My Profile',
          tabBarIcon: ({ color, size }) => (
            <MaterialCommunityIcons name="account" color={color} size={size} />
          ),
        }}
      />*/}
    </Tab.Navigator>
  );
}

const App = function(){
  return (

    <NavigationContainer>
```

```
      <PaperProvider theme={theme}>
        <MyTabs />
      </PaperProvider>
    </NavigationContainer>
  );
};
});
```

7.2.4 Home Screen

To begin the implementation of the Home Screen, we designed the communication architecture between the Frontend and Backend, summarised in Figure 28. The arrows represent connections and logic flow.

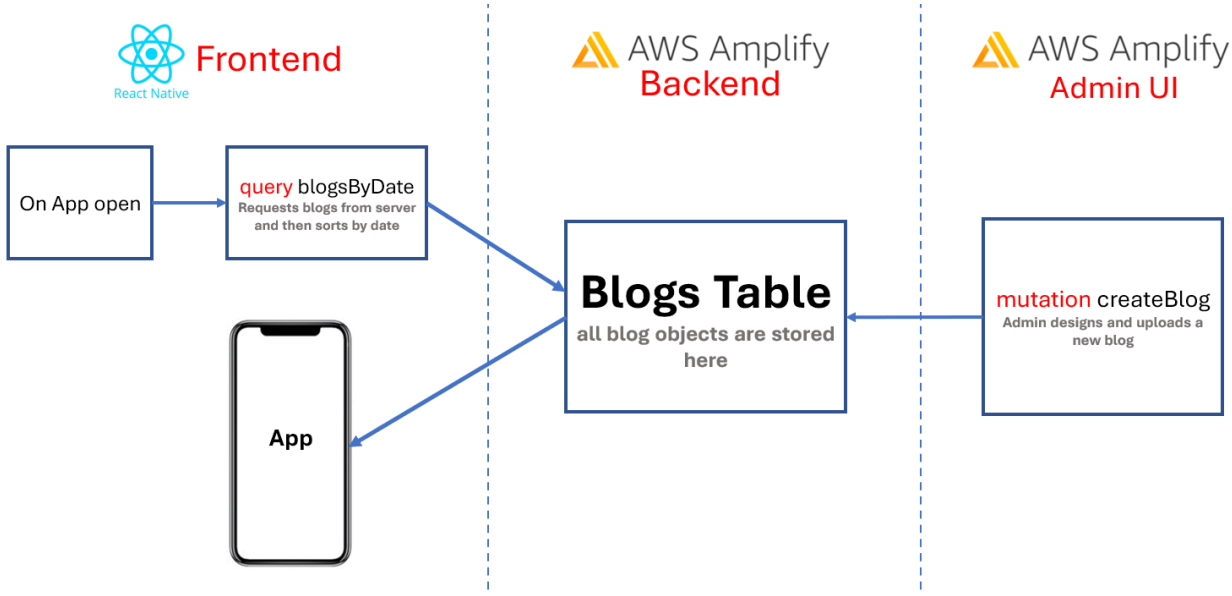

Figure 28: Communication architecture between the frontend and backend for the Home Screen

The GraphQL schema used for communicating information regarding each Blog/Article is as follows:

```
type BlogNew @model
@key(name: "byDate", fields: ["area", "createdAt"], queryField: "blogsByDate")
{
  id: ID!
  title: String!
  content: String!
  description: String!
  date: String!
  image: String
  createdAt: AWSDateTime
  updatedAt: AWSDateTime
  area: String!
}
```

`BlogNew` is the name of the schema, the line : `@key(name: "byDate", fields: ["area", "createdAt"], queryField: "blogsByDate")` allows us to create a query called "blogsByDate" which grabs the list of blogs from the server, filter the blogs by area and then sort such that the newest articles are at the top of the list. The `id` field is a default field, allocating a unique id for each blog. The `title`, `content`, `description`, `date` and `image` are used to display the content regarding each blog on the app.
Following the design of this architecture, and the design shown in Figure 23, we wrote the code to implement all the features specified up until now. The full code for this screen can be found in the Appendix (Appendix: App Code, HomeScreen.js), and includes thoroughly detailed comments that explain the role of each section.

7.2.5 Chat Room

Identical to the Home Screen, we first began the implementation by designing the communication architecture between the Frontend and the Backend, shown in Figure 29. The red arrows represent connections from the frontend to the Agora.io backend, and the dark blue represent connection bewteen the frontend and the AWS Amplify backend.

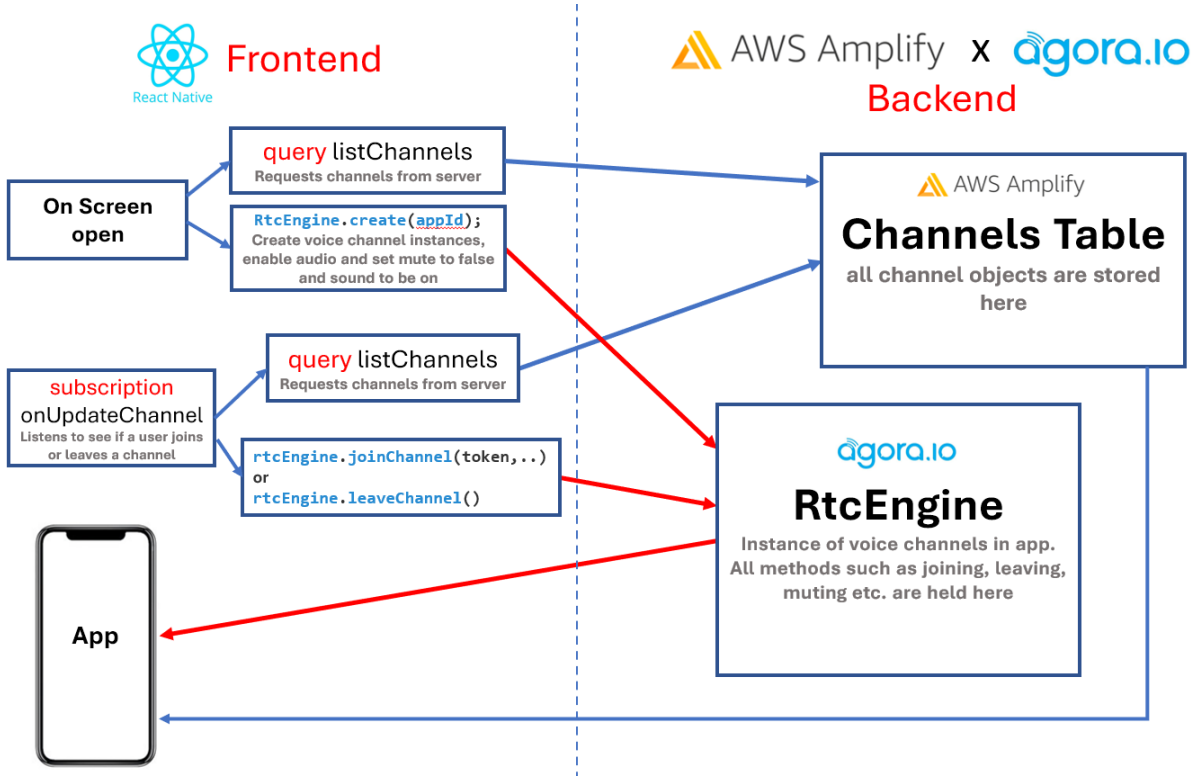

Figure 29: Communication architecture between the frontend and backend for the Chat Room

The Schema used for this section is as follows:

```
type Channel
  @model
  {
    id: ID!
    value: ID!
    label: String!
```

```
    appId: String!
    token: String!
    users: [String!]!
  }
```

The `label` field will contain the channel name, `appId` will hold the Agora.io appId although in hindsight this is not necessary and can be removed later, `token` will contain the unique token for the channel and will be used when joining a voice channel through Agora.io, and the `users` holds the names of all the users that are currently inside the channel.

The code for this screen is also included in the Appendix (Appendix: App Code, ChatRoom.js), it contains detailed comments that can be used later by any developers working on the project to update or improve upon existing code.

7.2.6 Message Us

The communication architecture is shown below in Figure 30:

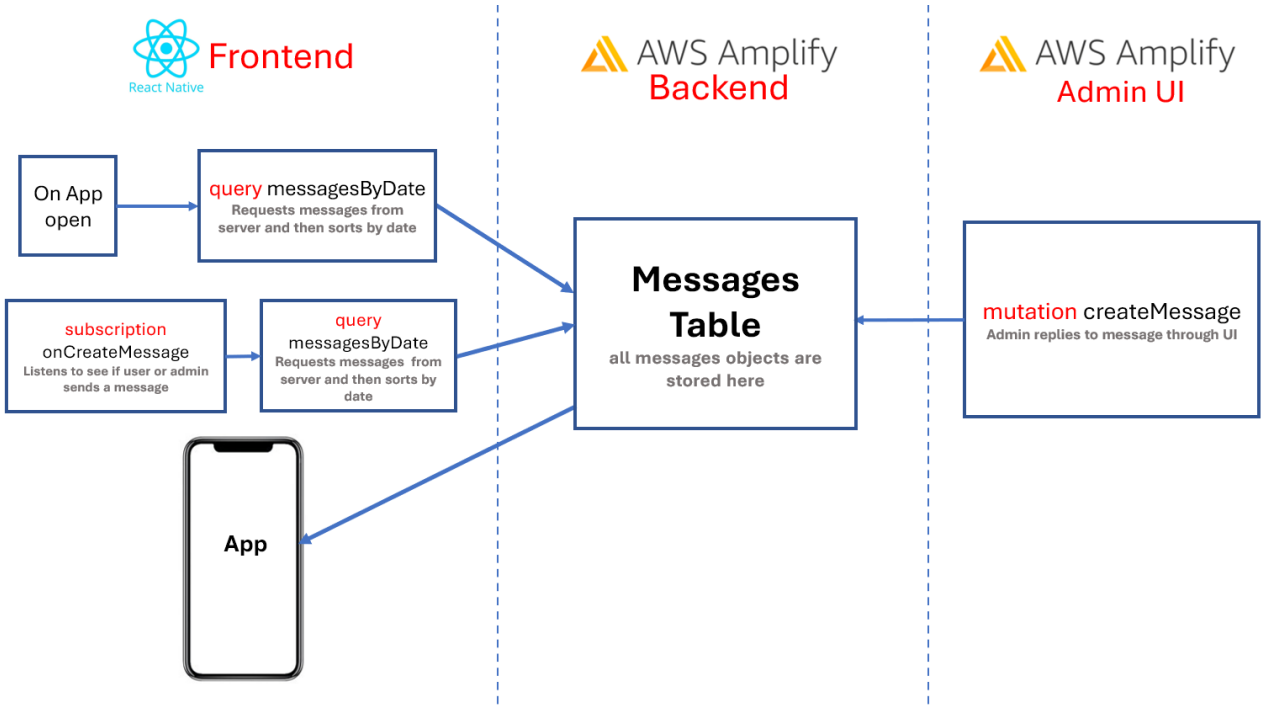

Figure 30: Communication architecture between the frontend and backend for the Message Us screen

The schema used for communicating messages is as follows:

```
type Message
  @model
  @key(name: "byChannelID", fields: ["channelID", "createdAt"], queryField: "
    messagesByChannelID")
  {
  id: ID!
  channelID: ID!
  author: String!
  body: String!
```

```
  createdAt: AWSDateTime
  updatedAt: AWSDateTime
}
```

Much like with `BlogsNew`, the key allows us to set up a query operation that filters the messages by channel ID, which is set to the current users id, this means that only the messages associated with the current user are shown. Once this is filtered, messages are sorted by time of sending, much like in all messaging services. The `author` field is set to either the user's id or to "admin", this allows us to differentiate between the two senders in the app. The `body` is message body. The code for this screen follows a tutorial closely[25], but custom styling and extra components are included to fit the design shown in Figure 26.

The code can be found again in the Appendix (Appendix: App Code, ContactScreen.js) and also contains detailed comments for later developers.

7.3 Maintaining and updating features in the app

7.3.1 Maintaining the App features: For Admins

We felt it was crucial to provide some sort of tutorial for admins on how to maintain the three main features of the app. To do this, we created a short tutorial videos that covers the uploading a new article to the Home Screen, the creation of a new voice channel, and replying to messages on the Message Us screen. This video tutorial can be found by navigating again to our github page for the project: `https://github.com/nimakarshenas/GreenwichTogether/blob/main/tutorial%20for%20admins.mp4`.

7.3.2 Building on Existing Code: For Developers

As mentioned before, the code for each feature and screen has been thouroughly commented for developers to understand the design of each page. On our github repository, We have also included a guide on all the steps and installations required in order to run the code on any machine, this will allow developers to swiftly begin appending to the existing code. `https://github.com/nimakarshenas/GreenwichTogether`

7.3.3 Taking to production

We used the Expo Cli App[26] to handle the development of the app. It was chosen for its easy-to-use command line interface that allowed us and will allow us to:

- Run the project server, view logs, open our app in a simulator.
- Publish our app and other assets and manage releasing them over the air.
- Build binaries (apk and ipa files) to be uploaded to the App Store and Play Store.

Once deemed to be ready to be published, this following in-depth tutorial will allow the app to be published on both the App Store and the Google Play Store: `https://docs.expo.io/distribution/uploading-apps/`.

7.3.4 Pricing

The two sources of expenses for the app come from the backend services used, Amazon Amplify AppSync data storage used for the GraphQL api and Agora.io. When calculating expenses we have made estimates on the number of users etc., these number can of course be easily adjusted once the council has a better idea of the app's traffic.

For the Home Screen feature, we estimate 10,000 monthly active users, each visiting the page an average of once per day, this leads to $10,000 \times 30 = 300,000$ query operations, each operation has an estimated size of 3KB[27], leading to the monthly expenses that are summarised in the table below:

Home Screen Pricing

Service (AppSync)	Cost Calculation
Query operation charges	0.3 million x $4.00 per million operations = **$1.20**
Data transfer charges	0.3 million x 3KB = 0.9 GB x $0.09 = **$0.081**
Total AppSync charges	**$1.20 + $0.081 = $1.281 = £0.92 per month**

Figure 31: Table showing the monthly expenses for the Home Screen

For the Message Us page, we estimate less monthly users (2,500), averaging 10 sent messages per month. Amazon estimates that 1KB of data transfer is required per message[27]. Finally, we estimate an average connection time of 3 minutes per message, leading to 30 minutes per user per month. These assumptions lead to the cost estimates shown in the table below:

Message Us Screen Pricing

Service	Cost Calculation
Query operation charges	2,500 users x 10 sent messages = 25,000 operations x $4.00 per million operations = **$0.1**
Data transfer charges	0.025 million operations x 1KB = 0.025 GB x $0.09 = **$0.002**
Real-time update charges	2,500 users x 10 received messages x $2.00 per million updates = **$0.05**
Connectivity charges	2,500 clients x 30 minutes x $0.08 per million connection-minutes = **$0.006**
Total AppSync charges	**$0.1 + $0.002 + $0.05 + $0.006 = $0.16 = £0.11 per month**

Figure 32: Table showing the monthly expenses for the Message Us Screen

Finally, for the Chat Room we estimate even fewer active monthly users (1,000), we estimate that they join channels 30 different times throughout the month, and with each time they join a channel this will require 5KB of data transfer (the data schema for the Channel contains more data than a Blog for example). We estimate that a user will spend on average 10 minutes inside a call, leading to 300 total minutes throughout the month. Real-time updates are performed when any new user joins a channel, we estimate this will happen once per minute for each user, leading to 300 real-time updates per user per month. Finally, we make the worst case assumption that the voice channel server will be running 24/7. All these assumptions lead to the monthly cost estimate shown in the table below:

Chat Room Pricing

Service	Cost Calculation
Query operation charges **(AppSync)**	1,000 users x 30 joined channels = 300,000 operations x $4.00 per million operations = **$1.20**
Data transfer charges **(AppSync)**	0.3 million operations x 5KB = 1.5 GB x $0.09 = **$0.135**
Real-time update charges **(AppSync)**	1,000 users x 300 channel changes x $2.00 per million updates = **$6.00**
Connectivity charges **(AppSync)**	1,000 users x 300 minutes x $0.08 per million connection-minutes = **$0.006**
Connectivity charges **(Agora.io)**	30 days x 24 hours x 60 minutes = 43,200 minutes – 10,000 free minutes = 33,200 minutes x $0.99/1,000 minutes = **$32.90**
Total AppSync charges	**1.20+0.135+6+0.006+32.90= $40.20 = £29.00 per month**

Figure 33: Table showing the monthly expenses for the Chat Room

The estimated total cost of operations is just over £30 per month for all of the features. To cover the costs of operation, the council can opt to include ads inside the app, this can later be added if needed.

7.3.5 Testing

Since the app has been designed for purely qualitative reasons i.e. user experience, it's difficult to quantitatively analyse its performance. The demo of the app shows the all the functionalities of the app in practice. This can be also be found on our GitHub repository. Throughout the design and implementation process we have kept inline with the design specifications outlined in Section 7.1.3 and made sure all of these had been met before moving onto the next feature.

7.3.6 Ethics and Sustainability

The main ethical and legal concern of the app comes in the form of Data Privacy. Apart from user's full name, email and phone number, we hold no other personal information. Messages between the user and admins are of course stored on the cloud, but we don't make use of this data, and AWS's high level of security[28] ensures even this data is safe. Finally conversations held in voice channels, are neither tracked or stored, furthermore, Agora.io encrypts audio ensuring that no-one can intercept these conversations.

In terms of sustainability, no raw materials are used for this app, it is purely software based and so there are no concerns regarding its ecological sustainability.

8 Project Management

8.1 Workflow and organisation

This project was conducted by 5 Third year students in the Electrical and Electronics Engineering department at Imperial College London. We had not worked together on a project as a team previously, but some of us had worked together on other assignments before.

8.1.1 Meeting record and minutes

During the first two weeks, we had a lot of meetings, with client, supervisor and just as a group to work through the first issues together. After that, we set up 2 weekly group meetings, on Monday and Wednesday, and a third one on Friday with the client. Every two weeks we included our supervisor in the Wednesday meeting. The structure of every meeting was very similar: each member went through the work they had been assigned and discussed it with the rest of the group. After that, we did an overview of where the project was heading, what were the next tasks, and divided them amongst the members. Meeting minutes were recorded in every meeting to keep track of the evolution of the project, and be able to go back a few weeks if we lost sight of the bigger picture. These minutes were very helpful when putting everything together at the end of the project. We have attached the meeting minutes in the appendix.

During the first week of the project, we came up with a timeline, which we respected until the end. The only thing we modified is that we first focused on the final presentation and did the report and the end, in the timeline we had planned to do it the other way, but we didn't know the timings of deadlines yet. The timeline has been included in the appendix, together with a Gantt Chart that complemented it. The most important part of the timeline is that research had to be completed by week 6 of the project. In other words, we had to find our solution before week 6 of the project. This allowed us to spend the last weeks working on deliverables and polishing the details and weak spots of our recommendation, instead of trying to fit in more information in the last minute. We met all the deadlines without any issue and leveraged our skills to divide the work between the team members. The organisation of the project in general has been very successful, we kept everything very structured, and were always conscious of what the next steps would be. Communication between team members has been flawless also.

The only thing that could have been improved about the project is the acquisition of the Huawei 5G cpe pro router, which we did not get because the supplier took more than 2 weeks. This would have provided us with the opportunity to perform crucial tests that would validate the speeds of the router in different parts of a house. We could have started the process earlier, but we did not really have the opportunity to go through the bureaucracy any quicker, so this is the only aspect to be improved of the project.

8.2 Tests

As we mentioned in the previous part, we couldn't realise the tests because we could not get the Router, but in this part we will detail the tests we wanted to complete and all the information we wanted to retrieve.
The main characteristics to test are:

- Download speed
- Upload speed
- Ping
- Jitter

Upload and download speed are self explanatory, the Ping lets you know how long it takes for a packet of data to travel from your computer to a server on the internet and back, and Jitter is defined as a variation in the delay of received packets. Speeds are measured in Mbit/s and Ping and Jitter in ms. To make sure the test results are reliable and to see the variation of our results we planned to do the following:

- Realise the tests in different parts of the house
- Realise the tests in different parts of the day
- Repeat every test 3 times and if no anomalies are seen, take the average for each measurement

Upload and download speed tests can be done by respectively uploading or downloading a file of a given size from the internet, but this method is not too precise so we planned on using the standardised tests which can be accessed in `speedtest.net`.

8.3 Environment

We have kept the environment at the forefront all the decisions we made about the technologies being used and the implementation plan. We decided to reuse the routers acquired in phase 2 for phase 3, so we won't have waste between phases and all the hardware can be reused. There is no need for big infrastructure to be placed for this project in itself, it is in some way "self-sustaining", there is no drilling, no digging up roads, and no damaging the ecosystem. We will just use the infrastructure which is already in place (Current 5G coverage and antennae) and will adapt to the changes (development of a more dense 5G antennae network).

8.4 Ethics

We are targeting a major social problem: access to information and opportunities. We are doing so only in the benefit of those in need, not hindering the access to information of any other collective. Efforts have been made in phase 1 to inform everyone and make sure that no one is left behind in joining the initiative and making the most out of it. Individualised communication are planned to reach even the least connected people. The app also will play a big role in making sure everyone is on board.

9 Appendices

9.1 Appendix: App code

HomeScreen.js

```
import React, { useState, useEffect} from "react";
import { Text, StyleSheet, View, FlatList, Image, TouchableOpacity, Modal, Pressable,
    ScrollView } from "react-native";
import {API, graphqlOperation } from 'aws-amplify';
import { Auth } from 'aws-amplify';
import  {blogsByDate } from '../graphql/queries';
import '@aws-amplify/pubsub';




const HomeScreen = () => {


    //state for storing result of api pull request for list of blog objects
    const [blogs, setBlogs] = useState([]);
    //state for controlling whether Modal JSX element is visible
    const [modalVisible, setModalVisible] = useState(false);
    //when item in List is pressed, this state points to the Blog object that has been
    pressed
    const [currentmodal, setCurrentmodal] = useState([]);

    /*async function signOut() {
        try {
            await Auth.signOut();
        } catch (error) {
            console.log('error signing out: ', error);
        }
    }*/

    /*get request from api and sort by date - Note:
    This will run only when the page opens*/
    useEffect(() => {
        API
          .graphql(graphqlOperation(blogsByDate, {
            area:"GreenwichWest", /*The area can be changed to be a user chosen variable
            if app is extended to more regions*/
            sortDirection: 'DESC'
          }))
          .then((response) => {
            const items = response?.data?.blogsByDate?.items;
            if (items) {
              setBlogs(items);  //update state holding the array of BlogNew objects
            }
          })
      }, []);



    // ------------- RENDERING GOES HERE ----------------------
```

```
return(

<View style={styles.mainContainer}>
    {/* MODAL setup --- when modalVisible is set to TRUE,
    whatever is embedded in the Modal JSX element is rendered
    on top of all other elements, acts as a pop-up */}
    <View style={styles.modalWrap}>
    <Modal
        animationType="slide"
        transparent={true}
        visible={modalVisible}
        onRequestClose={() => {
        Alert.alert("Modal has been closed.");
        }}>
        <View  style={styles.modalView}>
            <ScrollView style={{borderBottomWidth:0.5, borderColor:"#c20027"}}>
                {/* Image at the top of the page */}
                <Image
                    style={{ height:200, width:'100%',
                            borderRadius:5,padding:5,
                            borderWidth:2, borderColor:"#980e0e"}}
                    source={{uri: currentmodal.image}}>

                </Image>
                <View style={{borderBottomWidth:0.5,
                            borderColor:"#c20027", marginBottom:5}}>
                    <Text style={styles.titleModal}>
                        {currentmodal.title}
                    </Text>
                </View>
                <Text style={styles.descriptionModal}>
                    {currentmodal.description}
                </Text>
                <Text style={styles.contentModal}>
                    {currentmodal.content}
                </Text>
            </ScrollView>
            {/* Close Button */}
            <Pressable
                style={styles.button}
                onPress={()=>{setModalVisible(!modalVisible)}}>
                <Text style={{color:"#c20027", fontWeight:"bold"}}>Close</Text>
            </Pressable>
        </View>


    </Modal>
    </View>

    {/* This is what can be seen when the page first opens -- MODAL NOT OPEN */}
    <View style={styles.mainContainer}>
        {/* -------- HEADER ------------*/}
        <View style={{borderBottomWidth:0.5, borderColor:"#c20027"}}>
        <Image
        source={{uri:"https://www.london-works.com/sites/default/files/supporter-
logos/royal-borough-greenwich.png"}}
        style={{height:120, width:160, alignSelf:"center", paddingTop:-100}}>

        </Image>
        </View>
```

```
            {/* List of Tiles*/}
            <FlatList data={blogs}
                keyExtractor={(blog)=> blog.id}
                renderItem = {({item}) => {
                    return(

                        <View style={styles.centeredView}>
                        <View style={{marginTop:10, marginLeft:20, flexDirection:'row'}}>
                            <Text style={{ fontWeight:"600", fontSize:18, color:"#c20027"
    }}
                            >{item.date}</Text>
                            <View style={{borderBottomWidth:1, borderColor:"#c20027",
    flex:1,
                                    marginRight:10, marginBottom:8, marginLeft:5}}>
                            </View>
                        </View>
                        <TouchableOpacity style={styles.tileContainer}
                            onPress={() => {setCurrentmodal(item);
                            setModalVisible(!modalVisible)}}>
                            <View style={{borderBottomWidth:0.5, borderColor:'white'}}>
                            <Text style={styles.title}>{item.title}</Text>
                            </View>
                            <View style={{flexDirection:'row', marginTop:12}}>
                            <View style={styles.textInTileContainer}>
                                <Text style={styles.description}>{item.description}</Text
    >
                            </View>
                            <View style={styles.imageInTileContainer}>
                                <Image
                                    style={{flex:1, borderRadius:5}}
                                    source={{uri: item.image}}>

                                </Image>
                            </View>
                            </View>

                        </TouchableOpacity>
                        </View>
                    );
                }
            }/>
        </View>
        {/* CAN ADD A SIGN OUT BUTTON IF NEEDED*/}
        {/*<Button
            title="Sign out"
            onPress={signOut}
        />*/}
    </View>
    );
};


//--------------- STYLING GOES HERE ------------------
const styles = StyleSheet.create({
    modalView: {
        flex:1,
        overflow: 'scroll',
        marginVertical:100,
        marginHorizontal:20,
        backgroundColor: "white",
        borderRadius: 20,
```

```
        padding: 10,
        borderColor:"#c20027",
        borderWidth:5,
        alignContent:'stretch',
    },
    modalWrap:{
        shadowOpacity: 1,
        shadowRadius: 20,
        elevation: 15,
    },
    mainContainer:{
        flex:1,
        paddingTop:15,
        backgroundColor: 'white',

    },
    tileContainer:{
        backgroundColor: '#c20027',
        borderRadius: 15,
        margin:15,
        paddingVertical:15,
        paddingHorizontal:15,
        shadowOpacity:0.5,
        shadowRadius:5,
    },
    textInTileContainer:{
        flex:2,
        marginRight:10,
    },
    imageInTileContainer:{
        flex:1,
        marginLeft:5,
        shadowOpacity:0.5, shadowRadius:5,
        borderRadius:5,
        marginTop:5
    },
    title:{
        fontSize:24,
        fontWeight:'bold',
        color:'white',
        paddingBottom:5,
    },
    description:{
        paddingTop:5,
        fontSize:16,
        fontWeight:'bold',
        color:'white'
    },
    titleModal:{
        marginTop:15,
        paddingBottom:10,
        fontSize:28,
        fontWeight:'bold',
    },
    descriptionModal:{
        paddingTop:10,
        fontSize:18,
        fontWeight:'bold',
    },
    contentModal:{
        paddingTop:10,
```

```
        fontSize:16,
    },
    button: {
        marginTop:10,
        borderRadius: 20,
        padding: 10,
        elevation: 2,
        alignSelf:'center'
    },
})

export default HomeScreen;
```

ChatRoom.js

```
import * as React from "react";
import {useEffect, useState} from "react";
import { Text, StyleSheet, View, Image} from "react-native";
import { ScrollView, TouchableOpacity } from "react-native-gesture-handler";
import DropDownPicker from 'react-native-dropdown-picker';
import API, { graphqlOperation } from '@aws-amplify/api';
import  {listChannels } from '../graphql/queries';
import { Button } from 'react-native-elements';
import { Avatar } from 'react-native-paper';
import { List } from 'react-native-paper';
import { Auth } from 'aws-amplify';
import { onUpdateChannel } from "../graphql/subscriptions";
import { updateChannel } from "../graphql/mutations";
import RtcEngine from "react-native-agora";


/* Android must ask permission to use audio */
export const useRequestAudio = () => {
    if (Platform.OS === 'android') {
      try {
        const granted = await PermissionsAndroid.requestMultiple([
          PermissionsAndroid.PERMISSIONS.RECORD_AUDIO,
        ]);
        if (
          granted['android.permission.RECORD_AUDIO'] ===
          PermissionsAndroid.RESULTS.GRANTED
        ) {
          console.log('You can use the mic');
        } else {
          console.log('Permission denied');
        }
      } catch (err) {
        console.warn(err);
      }
    }
};


/* destroy rtcengine insstance when cleaning up*/
```

```
const destroyAgoraEngine = useCallback(async () => {
  await rtcEngine.destroy();
}, []);



const ChatRoom = () => {
  /* appId taken from Agora.io console*/
  const appId = "PUT_APPID_HERE";

  /* muteState set to the name of the icon, this could instead be a
  true/false but this made things easier changing between the mute
  and unmute icons when pressed.
  Note: "microphone" = unmute, "microphone-slash = mute"*/
  const [muteState, setMuteState] = useState("microphone");

  /* much like the muteState, md-volume-high = sound on,
  md-volume-mute = sound off*/
  const [soundState, setSoundState] = useState("md-volume-high");

  /* Button text for the join/leave button, it is also used to
  keep track of the users join status*/
  const [buttonText, setButtonText] = useState("Join");

  /*open state for dropdownpicker JSX element*/
  const [open, setOpen] = useState(false);

  /* Here we store all the infor regarding a channel, it is an
  array of objects with the following form:
  channelInfo:{
    appId: {string} ---- appId for agora.io
    createdAt: {AWSdatetime} ----- when the channel was created
    id: {string} ------ unique id for each channel, used for updateChannel mutation
    label:{string} ---- What is seen on the screen of the app for channel name
    value:{string} ---- This is used for DropDownPicker Element, can be the same as label
    updatedAt:{string} --- when channel was last updated = when user last joined/left a
    channel
    token:{string} --- unique channel token, used for joining channel in Agora
    users:{array[string]} --- list of user names inside ther channel
  }*/
  const [channelInfo, setChannelInfo] = useState([]);

  /* current selected channel in the dropdown selector*/
  const [selectedChannel, setSelectedChannel] = useState("");

  /* store the users full name so it can be displayed in the channel*/
  const [userFullName, setUserFullName] = useState({});

  /* What channel the user has currently joined*/
  const [joinedChannel, setJoinedChannel] = useState("");

  /* change the color of the users full name when shown on the dropdown
  that displays all the names of the users that have currently joined a
  specific channel*/
  const [muteColor, setMuteColor] = useState("#c20027");



  /* ----------- HELPER FUNCTIONS -------------*/
```

```
/* This is a simple helper function that finds the index of an
object within an array of a specific 'value' attribute */
const findindex = (ARR_OF_OBJS, CHAN_TO_FIND) => {
  for(let i=0 ; i < ARR_OF_OBJS.length ; i++ ){
    if(ARR_OF_OBJS[i].value === CHAN_TO_FIND){
      return i;
    }
  }
  return -1;
}

/* This essentially removes a user from a channel and returns a copy
of the array holding the objects */
const removeItem = (ARR_OF_OBJS, USER_TO_REMOVE, IDX_OF_OBJ) => {
  const FIN_ARR = []
  for(let i=0; i<ARR_OF_OBJS[IDX_OF_OBJ].users.length; i++){
    if(ARR_OF_OBJS[IDX_OF_OBJ].users[i] !== USER_TO_REMOVE){
      FIN_ARR.push(ARR_OF_OBJS[IDX_OF_OBJ].users[i])
    }
  }
  ARR_OF_OBJS[IDX_OF_OBJ].users = FIN_ARR;
  const copyArr = ARR_OF_OBJS;
  return copyArr;
}


/* This is a helper function for handleJoinPress(), we begin by finding
the index inside the array of the channel that we have joined, we then
remove the user(us) from the channel's user attribute(array of names),
and then update the Channel information (with the user removed) through
a graphql mutation, this is so that the user leaving is visible to everyone
using the app. We also leave the agora.io channel*/
const leaveChannel = async () => {
  const idx = findindex(channelInfo, joinedChannel);
  const newChannelInfo = removeItem(channelInfo, userFullName, idx);
  const forInput = newChannelInfo[idx];

  await rtcEngine.leaveChannel();

  try {
    await API.graphql(graphqlOperation(updateChannel,
      {input: { id: forInput.id, users: forInput.users}}))
  } catch (error) {
    console.warn(error);
  }
  setButtonText("Join")
}

/* This function operates much like leaveChannel(), but instead we join
the user to the channel both through the amplify backend and the agora.io
backend*/
const joinChannel = async () =>{
  const idx = findindex(channelInfo,selectedChannel);
  const forInput = list[idx];

  await rtcEngine.joinChannel(forInput.token, selectedChannel, null, 0);

  /* add userFullName(us) to the list of users inside the channel we
  have decided to join */
```

```
    const list = channelInfo.map(item =>
      item.value === selectedChannel
      ? {...item, users: [...item.users, userFullName]}
      : item );



    try {
      await API.graphql(graphqlOperation(updateChannel,
        {input: { id:forInput.id, users: forInput.users}}))
    } catch (error) {
      console.warn(error);
    }
    /* change the button text and the joined channel the selected channel*/
    setButtonText("Leave");
    setJoinedChannel(selectedChannel);

  };


  /* Function for obtaining initials for Avatar display*/
  const getInitials = (name) =>{
    const fullName = name.split(' ');
    const initials = fullName.shift().charAt(0) + fullName.pop().charAt(0);
    return initials;
  }



  /* ----- ALL BUTTON CONTROLS GO HERE ---- */

  /* This function is called when the join/leave button is pressed */
  const handleJoinPress = () => {
    if(buttonText==="Join" && selectedChannel != ""){
      joinChannel();
    }
    else{
      leaveChannel();
    }
  }

  /* first mute from agora, change the icon and also the colour of the text*/
  const handleMutePress = () =>{
    if (muteState==="microphone"){
      await rtcEngine.muteLocalAudioStream(true);
      setMuteState("microphone-slash")
      setMuteColor("#ECB2BE")
    }
    else{
      await rtcEngine.muteLocalAudioStream(true);
      setMuteState("microphone")
      setMuteColor("#c20007")
    }
  }

  /* similar process to handleMutePress*/
  const handleSoundPress = () =>{
    if (soundState==="md-volume-high"){
      rtcEngine.setEnableSpeakerphone(false);
      setSoundState("md-volume-mute")
    }
```

```
    else{
      rtcEngine.setEnableSpeakerphone(true);
      setSoundState("md-volume-high")
    }
  }




  /* ----- useEffect functions go here ---------*/

  /* when page first opens we request audio and pull from the Amplify backend
  the current, live information regarding the channels like what users are inside the
  channel etc. */
  useEffect(() => {
    useRequestAudio();
    /* set user info first */
    Auth.currentUserInfo().then((userInfo) => {
      const firstname = userInfo.attributes.name;
      const secondname = userInfo.attributes.family_name;
      setUserFullName(firstname + " " + secondname)
    })
    API
    .graphql(graphqlOperation(listChannels))
    .then((response) => {
      const items = response.data?.listChannels?.items;

      if (items) {
        setChannelInfo(items);

      }
    });

  }, [])

  useEffect(() => {

    const subscription = API
      .graphql(graphqlOperation(onUpdateChannel))
      .subscribe({
        //next acts as the callback for the listener
        next: (event) => {
          /* event has the following form:
          {
            provider: { ... }, // this contains AppSync subscription metadata
            value: {
              data: {
                onUpdateMessage: {
                  appId: {string}
                  createdAt: {AWSdatetime}
                  id: {string}
                  label:{string}
                  value:{string}
                  updatedAt:{string}
                  token:{string}
                  users:{array[string]}
                }
              }
            }
        }*/
```

```
        const idx = findindex(channelInfo, joinedChannel);
        var channInfoCopy = channelInfo;
        channInfoCopy[idx] = event.value.data.onUpdateChannel;
        setChannelInfo(channInfoCopy);
      }
    });
    /* return acts as the clean up method for the effect, the function is essentially
  called
    as the effect is called again or is unmounting */
  return () => {
    subscription.unsubscribe();
  }
}, [channelInfo] /* */);

useEffect(() => {
  rtcEngine = await RtcEngine.create(appId);

  await rtcEngine.enableAudio();
  await rtcEngine.muteLocalAudioStream(false);
  await rtcEngine.setEnableSpeakerphone(true);


  return () => {
    destroyAgoraEngine();
  };
}, []);

/* ---------------------------------- ALL RENDERING GOES HERE
  ----------------------------------------*/
return (

  <View style={{paddingTop:30, backgroundColor:'white', flex:1, justifyContent:'space-
  between'}}>


  <View style={{flexDirection:'row', justifyContent:'space-evenly',
  borderBottomWidth:0.5, borderColor:'#c20027' }}>
    <Image
      source={{uri:"https://www.london-works.com/sites/default/files/supporter-logos/
  royal-borough-greenwich.png"}}
      style={{height:120, width:160,
      alignSelf:"flex-start", marginRight:7}}>
    </Image>
    <Text
      style={{fontSize:35, color:"#c20027", fontWeight:"200",
      textAlign:"left", alignSelf:"flex-end", paddingBottom:15, marginLeft:7}}>
```

```
        Chat{"\n"}Rooms
      </Text>
    </View>

    {/* MAIN BODY */}
    <View style={{flexDirection:'row', borderBottomWidth:0.5,
    borderColor:'#c20027', paddingTop:10}}>

    <DropDownPicker

      listItemContainerStyle={{borderBottomWidth:0.5, borderColor:'#c20027',
                               height:50, marginHorizontal:7}}
      dropDownContainerStyle={{marginLeft:7, elevation:999,
                              borderColor:'#c20027', borderWidth:2}}
      value={selectedChannel}
      setValue={setSelectedChannel}
      open={open}
      setOpen={setOpen}
      placeholder="Choose your room!"
      items={channelInfo}
      containerStyle={{
        margin:5,
        width:"80%",
        marginLeft:7
      }}
      style={{
        height:60,
        marginLeft:7,
        marginBottom:10,
        borderRadius:15,
        borderWidth:2,
        borderColor:"#c20027",
        rippleColor:"#c20027",
        underlayColor:"#c20027"
      }}

      ArrowDownIconComponent={({style}) => <List.Icon icon="chevron-down" style={style
    }/>}
      ArrowUpIconComponent={({style}) => <List.Icon icon="chevron-up" style={style}/>}
      tickIconStyle={{color:"#c20027"}}
      arrowIconStyle={{color:"#c20027"}}
      textStyle={{fontSize:20, color:"#c20027", fontWeight:"600"}}
      bottomOffset={100}

    />
    <TouchableOpacity style={styles.joinButton}
      onPress={handleJoinPress}>
      <Text style={styles.joinButtonText}>{buttonText}</Text>
    </TouchableOpacity>
    </View>

    {/*list containing all the channel info */}
    <ScrollView style={{zIndex:-5}}>
      {channelInfo.map((channel) => {
        return(
          <List.Accordion
            title={channel.label}
            description="press to expand"
            descriptionStyle={styles.accordionHeaderDescription}
            style={styles.accordionHeader}
            titleStyle={styles.accordionHeaderText}
```

```
            left={props => <List.Icon {...props} icon="account-group" color='white' />}
            right={props => <List.Icon {...props}  icon="chevron-down" color='white'/>}
          >
          {/* display all the users inside the channel*/}
          {channel.users.length !==0 ? channel.users.map((user)=>{
            return(
              <List.Item title={user}
              titleStyle={{fontSize:26, paddingBottom:10,
                           fontWeight:"600", color: muteColor }}
              style={{marginHorizontal:20, height:65,
                      borderBottomWidth:1, borderBottomColor:'#c20027'}}
              left={props =>
                <Avatar.Text {...props}
                  size={40}
                  color='#f6e3ba'
                  label={getInitials(userFullName)}
                  />}
              />
            );
          }) : <List.Item title="There are no users in this channel"
              titleStyle={styles.noUsers} />
          }
          </List.Accordion>
        );
      })}
    </ScrollView>

    {/*----- USER CONTROLS GO HERE -----*/}
    <View style={styles.toolbarContainer}>
      <Button
        style={styles.toolbarButtonLeft}
        type="clear"
        icon={{
          type:"font-awesome",
          name: muteState,
          size: 35,
          color: "white"
        }}
        onPress={handleMutePress}
      />
      <Button
        style={styles.toolbarButtonRight}
        type="clear"
        icon={{
          type:"ionicon",
          name: soundState,
          size: 40,
          color: "white"
        }}
        onPress={handleSoundPress}
      />

    </View>



    </View>
    );
};
```

```
/* ----------------- STYLING ------------------- */
const styles = StyleSheet.create({
  joinButton:{
    flexDirection:'column',
    justifyContent:'center',
    width:50,
    marginLeft:10,
    marginTop:5
  },
  joinButtonText:{
    color:"#c20027",
    fontSize:16,
    fontWeight:"600",
    alignSelf:'center',
    paddingTop:20

  },
  text: {
    alignSelf:'center',
    marginTop: 50,
    fontSize: 30
  },
  channelContainer:{
    flex:1,
    marginLeft:20,

  },
  channelHeader: {
    borderRadius:30,
    marginBottom:15,
    height:60,
    flex:1,
    flexDirection:'row',
    backgroundColor:"#c20027",
  },
  channelHeaderText:{
    color:'white',
    fontWeight:"100",
    fontSize:24,
    paddingLeft:20,
    paddingTop:15
  },
  channelUser:{
    height:50,
    flexDirection:'row'
  },
  accordionHeader:{
    borderRadius:15, margin:10, backgroundColor:"#c20027",
    height:70
  },
  accordionHeaderText:{
    color:'white',
    fontSize:18,
    fontWeight:"600"
  },
  accordionHeaderDescription:{
    color:"#efe9d9"
```

```
  },
  userTitle:{
    fontSize:26,
    paddingBottom:10,
    fontWeight:"600",
    color: '#c20027',
  },
  noUsers:{
    fontSize:18,
    paddingRight:30,
    fontWeight:"600",
    color: '#c20027'
  },
  toolbarContainer:{
    flexDirection:'row',
    justifyContent:'space-between',
    height:60,
    borderTopWidth:0.5,
    borderColor:'white',
    backgroundColor:'#c20027',
  },
  toolbarButtonLeft:{
    alignSelf:'flex-start',
    paddingTop:5,
    paddingLeft:15
  },
  toolbarButtonRight:{
    alignSelf:'flex-end',
    marginBottom:5,
    paddingRight:15
  },
  icon:{
    color:'#c20027'
  }
});

export default ChatRoom;
```

ContactScreen.js

```
import React, {useState, useEffect} from "react";
import { View, SafeAreaView, Text, StyleSheet, KeyboardAvoidingView, Keyboard,
    TouchableWithoutFeedback, Linking, Platform, VirtualizedList } from "react-native";
import { Input } from 'react-native-elements';
import { Button } from 'react-native-elements';
import API, { graphqlOperation } from '@aws-amplify/api';
import  {messagesByChannelId } from '../graphql/queries';
import '@aws-amplify/pubsub';
import { createMessage } from '../graphql/mutations';
import { onCreateMessage } from '../graphql/subscriptions';
import { Auth } from '@aws-amplify/auth';
```

```
const ContactScreen = () => {

  /*state containing the list of message
  object between the users*/
  /* Message object follows from GraphQL schema and has the following form,
  where the type of the object attribute is shown in between the curly braces:
  message: {
    id: {string}
    channelID: {string}
    author: {string}
    body: {string}
    createdAt: {AWSDateTime}
    updatedAt: {AWSDateTime} see Amplify documentation for info on how to handle
  } */
  const [messages, setMessages] = useState([]);

  //This state contains the content inside the "Type your message here" input
  const [messageBody, setMessageBody] = useState('');


  /* This contains the current users info that
  is derived from the authentication flow, the object has the following form:
  userInfo: attributes : {
    email: "XXXX@XXXXX.com",
    email_verified: true,
    family_name: "Karshenas",
    name: "Nima",
    phone_number: "+44XXXXXXXXX",
    phone_number_verified: false,
    sub: "819a8126-30e4-43ab-8b0b-8bb7f41c07d8",
  },
  id: "us-west-2:4e6ab2fc-aa46-4f0e-8a89-af5d7cfd98b5",
  username: "819a8126-30e4-43ab-8b0b-8bb7f41c07d8",
  }
  }*/
  const [userInfo, setUserInfo] = useState({});


  /* FUNCTIONS FOR VIRTUALISED LIST */
  const getItem = (data, index) => ({
    body: data[index].body,
    author: data[index].author
  });
  const getItemCount = (data) => data.length;



  /* This is the function called when the call button is pressed,
  pop-up with option to dial the number specified*/
  const dialCall = async () => {

    let phoneNumber = '';
    //This process has slightly different syntax in Android and iOS
    // Phone number currently set to council helpline
    if (Platform.OS === 'android') {
      phoneNumber = 'tel:${02088548888}';
    }
    else {
      phoneNumber = 'telprompt:${02088548888}';
    }
    //Open the prompt to call the number
```

```
  Linking.openURL(phoneNumber);
};

/* When the message input detects a change this function is called,
it then sets the messageBody state to what is currently inside the input*/
const handleChange = (body) => {
  setMessageBody(body)
};

/* This function is called when the send button is pressed,
We essentially create a graphQL mutation to upload the message to the server,
the channel id is set to the username which is essentially the user id,
the author attribute is the same, and finally the body of the message is
set to the messageBody state which contains the contents of the input box.
The .trim() method removes whitespace from each end of the string*/
const handleSubmit = async () => {
  const input = {
    channelID: userInfo.username,
    author: userInfo.username,
    body: messageBody.trim()
  };

  try {
    //When message is sent, the input box should become empty again
    setMessageBody('');
    //Upload the message to the server
    await API.graphql(graphqlOperation(createMessage, { input }))
  } catch (error) {
    console.warn(error);
  }
};

/* set the userInfo state when the page first mounts */
useEffect(() => {
  Auth.currentUserInfo().then((userInfo) => {
    setUserInfo(userInfo)
  })
}, [])

/* Queries the messages for this specific channel, sorts by date and stores
them in the 'messages' state variable. Note this effect is called each time
userInfo changes, this is just a safety measure incase it changes*/
useEffect(() => {

  API
    .graphql(graphqlOperation(messagesByChannelId, {
      channelID: userInfo.username,
      sortDirection: 'ASC'
    }))
    .then((response) => {
      console.log(response)
      const items = response?.data?.messagesByChannelID?.items;

      if (items) {
        setMessages(items);
      }
    })
}, [userInfo]);

/* When handleSubmit(){line 71} is called, it uploads the message to the server using
  the
```

```
createMessage graphQL mutation,
this method essentially adds a listener to the createMessage function, when this is
  called
in handleSubmit(), it pulls the new, updated list of messages and updates
the messages that can be seen on the UI.
Note: each time messages changes this method is called,
this happens when this page first opens and first set of messages queried from the
  server,
and then when the messages are updated in this subscription/listener*/
useEffect(() => {
  const subscription = API
    .graphql(graphqlOperation(onCreateMessage))
    .subscribe({
      //next acts as the callback for the listener
      next: (event) => {
        /* event has the following form:
        {
          provider: { ... }, // this contains AppSync subscription metadata
          value: {
            data: {
              onCreateMessage: {
                id: "374fdfe6-71e4-4e02-b41f-d9d78037c916",
                channelID: "1",
                author: "Dave",
                body: "This should be the new message",
                createdAt: "2020-08-01T04:43:58.335Z",
                updatedAt: "2020-08-01T04:43:58.335Z",
              }
            }
          }
      }*/
        setMessages([...messages, event.value.data.onCreateMessage]);
      }
    });
    /* return acts as the clean up method for the effect, the function is essentially
  called
    as the effect is called again or is unmounting */
  return () => {
    subscription.unsubscribe();
  }
}, [messages] /* */);









/*  RENDER SECTION */



return(
  <TouchableWithoutFeedback onPress={Keyboard.dismiss} accessible={false}>
  <View
    style={styles.mainContainer}
  >

  {/*---------- HEADER ----------*/}
```

```
<View style={styles.headerContainer}>
  <View style={styles.headerItemContainer}>
    <Text style={styles.headerText}>Call us instead</Text>
    <Button
        style={{alignSelf:'flex-end'}}
        type="clear"
        icon={{
          name: "call",
          size: 30,
          color: "white"
        }}
        onPress={dialCall}
        />
  </View>
</View>

{/* Container holding the messages*/}
<SafeAreaView style={{flex:1, width:'100%'}}>
<VirtualizedList
  getItem={getItem}
  getItemCount={getItemCount}
  initialNumToRender={15}
  data={messages}
  keyExtractor={messages=>messages.id}
  renderItem={({item})=>{ return(
    <View
    style={item.author == userInfo.username ? styles.messageMe : styles.messageYou}>
    {/* Change styling on whether the message is from the user or the admin */}
    <Text style={item.author == userInfo.username ? styles.messageMeText : styles.
messageYouText}>{item.body}</Text>
    </View>
  );}}/>
</SafeAreaView>

{/* COntainer holding the input box and send button */}
<KeyboardAvoidingView style={styles.inputContainer}
behavior="padding"> {/* This means that this container moves up with the keyboard
when it is opened  */}
<Input
  value={messageBody}
  multiline
  placeholder="Type your message here..."
  placeholderTextColor="white"
  containerStyle={{flex:5, paddingTop:10, marginBottom:-13,backgroundColor:'#c20027'
}}
  inputContainerStyle={{borderRadius:30, backgroundColor:'#c20027', borderWidth:1,
borderColor:'white'}}
  inputStyle={{padding:15, paddingTop:15, color:'white', fontSize:18, fontWeight:'300
'}}
  onChangeText={handleChange}
/>
<Button
  style={{alignSelf:'flex-end', flex:1, paddingTop:18, backgroundColor:'clear'}}
  type="clear"
  icon={{
    name: "send",
    size: 30,
    color: "white"
  }}
  onPress={handleSubmit}
  />
```

```
    </KeyboardAvoidingView>

    </View>
    </TouchableWithoutFeedback>

    );
};




/* -------------- STYLING GOES HERE ----------------- */
const styles = StyleSheet.create({

  mainContainer:{
    flex:1, justifyContent:'space-between', backgroundColor:'white', flexDirection:'
    column'
  },
  headerContainer:{
    zIndex:-5,
    height:90, width:'100%',  alignSelf:'flex-start', backgroundColor:'#c20027',
    justifyContent:'flex-end', borderBottomWidth:1, borderColor:'black'
  },
  headerItemContainer:{
    flexDirection:'row', justifyContent:'space-between'
  },
  messagesContainer:{
    flex:1, backgroundColor:'white'
  },
  messageYou: {
    marginLeft:5,
    alignSelf:"flex-start",
    marginTop:4,
    paddingHorizontal: 12,
    paddingVertical:12,
    maxWidth:250,
    backgroundColor:"#c20027",
    borderRadius:25,

  },
  messageYouText:{
    fontSize:16,
    fontWeight:"400",
    color:'white'
  },

  messageMe: {
    marginRight:5,
    marginTop:4,
    paddingHorizontal: 12,
    paddingVertical:12,
    maxWidth:250,
    borderRadius:25,
    alignSelf:"flex-end",
    backgroundColor:'white',
    color:"#c20027",
    borderWidth:1.5,
    borderColor:"#c20027",

  },
```

```
  messageMeText:{
    fontSize:16,
    fontWeight:"400",
    color:"#c20027"
  },

  inputContainer:{
    zIndex:-5,
    borderTopWidth:1,
    borderColor:'black',
    alignSelf:'flex-end',
    backgroundColor:'#c20027',
    flexDirection:'row'
  },

  headerText:{
    marginLeft:10,
    marginTop:5,
    alignSelf:'flex-start',
    fontSize:30,
    fontWeight:"400",
    color:'white'
  },

});

export default ContactScreen;
```

9.2 Appendix: Timeline

Week 1:

- First meeting with the client and clarification of project objectives and theme.
- Preliminary research: exhaustive list of all possible solutions without any filter.
- Focus on highly innovative solutions.

Week 2:

- Meeting with the supervisor to discuss initial ideas and potential solutions: fttp, fttc, dsl, public wifi, 5g network, satellite network, underground and sewer lines for fibre deployment.
- Discuss the initial ideas and ask further questions to client on the scope of the project, the target customer base and objectives.
- Eliminate the least feasible solutions by the end of the week.

Week 3:

- Start a draft of the final presentation.
- Start a draft of the final report.
- Choose the two main solutions we want to focus on and analyse further.

Week 4:

- Finalise what we want as content in the presentation (skeleton, not content).
- Finish abstract, introduction, and "soft part of the report".
- Start deep feasibility study of the financial + technical feasibility of the two chosen options.

Week 5:

- Finish deep feasibility study and start adding it to the report.
- All the research should be complete at the end of the week.
- Start deciding what we should include in the demo video and poster.

Week 6:

- Work on the slides.
- Finish the report.
- Do the demo video and poster.

Week 7:

- Finish the slides and last touches to the report.

Week 8:

- Prepare final presentation / rehearse.

References

1. https://www.government.se/496173/contentassets/afe9f1cfeaac4e39abcdd3b82d9bee5d/sweden-completely-connected-by-2025-eng.pdf
2. https://www.broadbandgenie.co.uk/broadband/help/fttp-fttc-fibre-broadband-guide
3. https://www-file.huawei.com/-/media/CORPORATE/PDF/News/4G-Wireless-Broadband-Industry-WHITE-PAPER.pdf?la=zh
4. https://venturebeat.com/2019/12/10/the-definitive-guide-to-5g-low-mid-and-high-band-speeds/
5. https://5g.co.uk/guides/5g-frequencies-in-the-uk-what-you-need-to-know/
6. https://www.ofcom.org.uk/__data/assets/pdf_file/0022/111883/enabling-5g-uk.pdf
7. https://www.youtube.com/watch?v=qjqYRHYLoWo&list=PLTv48TzNRhaKOPQ_nDTHzSDJK36AKuqaW&ab_channel=WirelessFutureWirelessFuture
8. https://www.telit.com/wp-content/uploads/2018/05/Telit_White_Paper_LTE_Evolution.pdf
9. J. Hoydis, C. Hoek, T. Wild and S. ten Brink, "Channel measurements for large antenna arrays," 2012 International Symposium on Wireless Communication Systems (ISWCS), 2012, pp. 811-815, doi: 10.1109/ISWCS.2012.6328480.
10. https://ma-mimo.ellintech.se/2018/05/17/3d-beamforming-is-that-massive-mimo/
11. https://www.strategyand.pwc.com/gx/en/insights/2019/making-5g-pay/making-5g-pay.pdf
12. https://rusi.org/sites/default/files/20200602_5g_cyber_security_final_web_copy.pdf
13. https://www.5gspaceappeal.org/the-appeal
14. https://www.bcg.com/publications/2021/accelerating-broadband-adoption-for-remote-education-low-income-households?linkId=118543272
15. http://www.three.co.uk/Discover/Network/Coverage
16. https://www.uswitch.com/broadband/guides/what-is-5G-home-broadband/
17. https://5g.co.uk/news/three's-continuing-its-mission-to-deliver-the-uk's-fastest-5g/5498/
18. https://www.bbc.co.uk/bitesize/guides/zr3yb82/revision/2
19. https://www.wavedirect.net/blog/mesh-networking-advantages-disadvantages
20. https://www.wavedirect.net/blog/mesh-networking-advantages-disadvantages
21. https://www.statista.com/statistics/1107269/average-number-connected-devices-uk-house/
22. https://www.amazon.co.uk/NETGEAR-Protection-Satellite-Extenders-Eliminate/dp/B07T4CJBDG/ref=psdc_949408031_t3_B08FT9BLQX?th=1
23. https://gs.statcounter.com/os-market-share/mobile/united-kingdom
24. https://docs.amplify.aws/lib/auth/getting-started/q/platform/js

25. https://medium.com/geekculture/building-a-chat-app-on-react-and-aws-amplify-part-2-b6a44158e7ac

26. https://docs.expo.io/workflow/expo-cli/

27. https://aws.amazon.com/appsync/pricing/

28. https://www.privoit.com/resources/is-aws-more-secure-than-on-premise

EEE Group 5 2020/21 Meeting Minutes

Meeting Week	DISCUSSIONS	ACTIONS
Week 1	**<u>Preliminary meeting with Client</u>** 1. Define project objectives and deliverables • Project founded by the Greenwich council, to think about how we can deliver better connectivity/devices/training with the aim of getting residents internet connectivity and covid-related information. 2. Establish regular cadence – weekly meetings on Fridays 3. Clarify resources available to us (findings by the council, research materials, budget, etc.)	**Arnau:** Read through council documents provided and do further research **Chidera** and **Nima**: Look at where in the world there is great connectivity, how they implemented it **Yun**: Social research about Greenwich (trends/demographics/challenges related to connectivity) **Seth**: Research the latest technologies being implemented and their respective costs
Week 1	**<u>Preliminary meeting with Supervisor</u>** 1. Aligned on project objectives and deliverables 2. Suggestions on further research: • Connectivity in Spain and Sweden – leaders in the European Union • Satellite (DSL) internet • Running fibre cables through tube (underground) lines • 4G/5G Home internet	**Arnau:** Research into Spain & Sweden connectivity models **Chidera:** Look into the tube option for fibre deployment **Nima**: Research into 5G **Yun**: Market research into internet providers in Greenwich, their products and offers **Seth**: Research and compare satellite (DSL) option
Week 2	**<u>Week 2 Client Meeting</u>** 1. Clarify project objectives: • Connectivity to the internet is the main issue – the higher the speed the better • Growth / long-term prospects not a priority • Woolwich is the most critical area • Fibre is currently being deployed in	**Arn**au: Draft project timeline, start draft for final report. Technical research into 5G Chidera: Look into repeaters for 5G Nima: Look for other apps that give a neighbourhood and community experience

	the region, but it would take more than two years for residents to get connectivity 2. Comments on Week 1 research: • Satellite has not been as reliable as cables • 5G option is interesting – need more research on whether it will suffer from blocking (concrete walls in large buildings). Are repeaters useful? • Public WiFi is a good idea that could be explored further • Connectivity app is a good suggestion	**Seth**: Look into public WiFi for a specific building like a council home. **Yun**: Look at 4G and its performance
Week 3	**<u>Week 3 Client Meeting</u>** 1. Progress update: • 5G solution seems viable. Three are eager to expand 5G network • The use of extenders / repeaters are not viable, however Mesh Networks seem promising • Complete 4G coverage in most of Greenwich – seems like a quick win solution • 5G infrastructure can be easily built onto existing 4G ones 2. Comments on Week 2 research: • Safety and security concerns related to 5G, in particular with Huawei devices • Compare 5G prices to existing Fibre plans • Do more research into mesh networks	**Arnau**: Look at Toronto example for mesh networks; ask department for funds to purchase router. Research into Huawei security concerns. **Chidera** & **Seth**: Understand and explain mesh networks (compare it to star); Look into cybersecurity in meshed networks **Nima**: Work on the app; Compare Huawei router to FTTP/FTTC subscriptions. **Yun**: Look into social aspect, fears/concerns related to 5G, how can we reach more people?
Week 4	**<u>Week 4 Client Meeting</u>** 1. Comments on Week 3 research: • Consider doing a mesh network architecture and use that as framework • Look at how much internet costs today when building the business case • Look into implementation/ installation of mesh networks in large buildings • 4G to 5G transition plan seems	**Arnau**: Start writing report. Help Nima with slides. Contact the stores. **Nima**: Keep working on app. Work with Arnau on building the storyline of the slides **Seth** & **Chidera**: Come up with floorplan and architecture example

	viable	**Yun**: Look at subsidising options and business case, and risk of 5G. Find template and software for leaflet.
Week 5	**<u>Week 5 Client Meeting</u>** 1. Comments on Week 4 research: • Focus on the message you want to sell • In leaflet, detail the problems we will give solutions to • In architecture example, will need to take into account lifts, fire exits, and stairs when setting up the mesh networks architecture. • Include the estimates of setting up and operating costs (maintenance and managing costs).	**All**: Work on leaflet contents and design **Chidera**: Look into Cisco network architecture **Seth**: Amend architecture example based on Client's comments
Week 6	**<u>Week 6 Client Meeting</u>** 1. Progress update: • Council to get in touch with ISP and see if they can expand their 5G coverage to Greenwich, with the bargaining power of being able to give them more customers. • Initial idea is to have a router like a Huawei, that can exist on its own. • Proposed solution can be scalable if we build mesh networks in buildings.	**All**: Finalise content and structure of report and presentation **Yun**: Design the presentation slides

Gantt Chart (Group 5)

Client: DG Cities
Supervisor: Dr. Javier Barria

Start date:
Mon, 26/4/21

TASK	START	END
Setting Project Objectives		
Defining the problem	26/4/21	27/4/21
Review client materials	27/4/21	30/4/21
Align objectives with client	30/4/21	1/5/21
Align objectives with supervisor	1/5/21	2/5/21
Planning The Research Design		
Setting hypotheses	29/4/21	30/4/21
Formulate research methodology	1/5/21	5/5/21
Finalising research design	5/5/21	7/5/21
Data Collection and Analysis		
Primary research	11/5/21	15/5/21
Data processing and analysis	16/5/21	19/5/21
Reformulating hypotheses	21/5/21	22/5/21
Develop shortlisted options	23/5/21	27/5/21
Realignment with client	27/5/21	28/5/21
Developing The Business Case		
Evaluating results and drawing conclusi	28/5/21	30/5/21
Create implementation plan	30/5/21	3/6/21
Develop risk mitigation strategy	3/6/21	4/6/21
Finalise leaflet design	28/5/21	4/6/21
Delivering The Final Recommendation		
Undergo financial analysis	4/6/21	6/6/21
Document the business case	6/6/21	11/6/21
Present recommendation to client	6/6/21	17/6/21
Finalise documentation for submission	17/6/21	25/6/21