

Abstract

This project is aimed at enabling migrants and refugees to seek help from the concerned federal/local government and social departments. The system is established to empower the incoming people to the UK to be able to convey their issues to the concerned authorities in an effective manner. The system features a translation module capable of translating the website content to different popular migrant languages, hence providing an interface for effective communication. The primary users will be able to chat (audio/video, text) with the councillors of the related departments. Due to the pandemic, local offices and officials need to maintain social distancing and the traditional walk-in interviews are no longer efficient as the risk of spreading the virus persists, hence the need for a digital system is required. A survey was also conducted to understand the issues faced by the migrant communities. The system is aimed to maintain a history of the queries registered on the system and their respective response from the authorities.

Acknowledgements

Firstly, I would like to thank my supervisor, Catherine Teehan, for her great support, valuable advice, and patience throughout the entire project. Her help and knowledge were crucial for my progress as I greatly improved my skills and knowledge and efficiently managed my time.

I would also like to express gratitude to my friends for their kindness, patience, and moral support. Finally, I am grateful to the participants in the survey for user testing. Without their help and contribution, I would not have been able to evaluate and develop this project to the required level.

Table of Figures

Figure 1 Histogram Showing English Proficiency	9
Figure 2 Histogram Showing migrants duration in the UK.	9
Figure 3 Histogram Showing migrants satisfaction rate of the social services.....	10
Figure 4 Histogram Showing rate of migrant difficulty in finding transport.....	10
Figure 5 Histogram Showing rate of the reasons for language barrier	11
Figure 6 Histogram Showing migrant's difficulty rate on finding GP.....	12
Figure 7 Histogram showing the rate of migrant's success in communicating with authorities.....	12
Figure 8 Histogram Showing issues faced in finding work/Job.....	13
Figure 9 Histogram Showing kind of problems migrants face when claiming social rights.....	13
Figure 10 Histogram Showing the degree Covid19 affects migrants.....	14
Figure 11 Histogram Showing the percentage of immigrant's viewpoint if web based services solves their issue.....	15
Figure 12 Refugee Cardiff Menu	16
Figure 13 Welsh Refugee Council	18
Figure 14 Sanctuary Home Page	19
Figure 15 Use Case Diagram	35
Figure 16 System Generated ERD	63
Figure 17 Sequence Diagram	65
Figure 18 Project Structure	71
Figure 19 Translation API linkage.....	72
Figure 20 Home Page Translation menu.....	72
Figure 21 Home Page translated in Spanish.	73
Figure 22 Emergency Chart Translated in Spanish	73
Figure 23 Django Super Admin login Interface	74
Figure 24 Django Super Admin Interface	75
Figure 25 Login Form for Admin and other users	76
Figure 26 Admin Dashboard	76
Figure 27 Add hospital Form	77
Figure 28 Query Creation Form	78
Figure 29 Query Approval Form.....	79
Figure 30 Request Assignment.....	79
Figure 31 Counsellor Dashboard.....	80
Figure 32 WebRTC diagram	81
Figure 33 Class based view for chat threads.....	82
Figure 34 text chat between a migrant and counsellor.....	83
Figure 35 Example Test Case.....	84

Table of Tables

Table 1 Django versus Flask	5
Table 2 pros and cons of the Refugee Cardiff website	17
Table 3 Pros and Cons of the Welsh Refugee Council website.....	18
Table 4 Pros and Cons of the Sanctuary (Welsh Government) website	19
Table 5 Use Case Benefits Analysis	40
Table 6 Example Test Case.....	84
Table 7 System Test Table.....	85

Contents

Abstract.....	i
Acknowledgements.....	ii
Table of Figures.....	iii
Table of Tables	iv
Contents.....	v
1 Introduction	1
1.1 Project Scope	2
1.2 Project Aims and Objectives	2
1.3 Approach.....	3
1.4 Technology stack.....	3
1.4.1 Backend Technology	3
1.4.2 Frontend Technology	5
1.4.3 Storage	6
1.4.4 Code Environment	6
1.4.5 Project Management	6
1.4.6 Software Testing	6
1.4.7 Google Tools.....	6
1.4.8 Code Configuration	6
2 Background	7
2.1 Survey.....	7
2.1.1 Survey Results:	8
2.1.2 Survey Infographics.....	9
2.2 Existing Systems	15
2.2.1 Refugee Cardiff	15
2.2.2 Welsh Refugee Council	17
2.2.3 Sanctuary (Welsh Government).....	18
3 Approach.....	19
3.1 Functional Requirements.....	19
3.2 Non-functional requirements	33
3.3 Use Case Diagram	34
3.4 Use Cases Benefits chart.....	36
3.5 Use case Descriptions	41

4	Implementation	62
4.1	System Diagrams.....	62
4.2	Entity Relationship Diagram (ERD).....	62
4.3	Sequence Diagram	65
4.4	Sequence Diagram Explanation	66
5	System Code Explanation.....	66
5.1	MTV	66
5.1.1	Model:.....	66
5.1.2	Template	67
5.1.3	View.....	69
5.2	Code Distribution	70
5.3	Translation Module	71
5.4	Admin Dashboard	74
5.5	Query Module	77
5.6	Chat Module.....	80
5.6.1	WebRTC.....	80
6	Evaluation and Results	83
6.1	Test Cases.....	83
7	Future Work	85
8	Conclusion	85
9	Reflection	86
10	References	88
11	Appendices.....	89
11.1	test cases.....	89

1 Introduction

There is an ongoing refugee crisis in various parts of the world mainly due to war. This crisis resulted in different developed countries hosting populations of people coming from other states. According to the UNHCR, there were approximately 133,094 refugees in the UK and out of these around 62,000 have pending asylum cases as well (UNHCR, 2019) . These people coming in from different countries encounter many problems as they take a significant amount of time adjusting to the new local culture. Effective communication between the refugees and the authorities remains a big issue as most of them are not able to speak or read English and most of the authorities are not able to effectively understand the problems of the migrants, as these migrants may be of diverse language origins and it becomes very hard to hire translators for a diverse spectrum of languages.

It is also observed that in a great number of instances these newcomers seek out very basic kind of information such as (Equality and Human Rights Commission , 2018):

- Where is the nearest hospital?
- Where is the state department?
- How can I apply for a residence?
- Which schools are giving admissions to migrants?
- Where can I get help regarding COVID'19 Standard Operating Procedures (SOPs)?

The pandemic has made life more difficult for everyone especially the refugees. The UK is experiencing one of the worst crises in its history due to the rapid spread of the COVID'19 Virus and its new variants that are challenging and pushing the medical resources to the maximum throughout (Lydia McMullan, 2021). The migrants during this time are facing much more difficulties than the regular communities in the UK. There are different timings and SOPs of different local departments, they have a hard time adjusting to the new culture as it is, and the pandemic lockdown and social distancing restrictions have made it hard for them to adjust properly. It takes migrants a long time to seek effective help from various state departments as the resources of the government are exhausted in trying to compete with the demands of the pandemic.

In this situation, the effectiveness of traditional walk-in queries is no longer feasible and there is a great need for an automated solution that will be able to cater for these needs. This project is greatly aimed to help the migrants in seeking effective help smartly and digitally. It has been seen that the entire world adopted the "Work from Home" model and it has been working great in many different places (UK GOV, 2021). Technology has empowered mankind to counter the pandemic in a far better way than our ancestors would have been able to do so. The whole concept of technology is that it exists to help human beings make a better and tolerant world for everyone, regardless of any cultural or ethnic boundaries.

1.1 Project Scope

The scope of this project evolved through various software engineering steps. The initial idea of the project went through the evolutionary cycle as more information was gathered and after the relevant processing of the information, proper decisions were made regarding the actual implementation of the project. Research conducted for this project was rather focused on the needs of the targeted community instead of the capabilities of the existing systems. This project considered the different geographic regions of the UK, hence the decisions taken, and the processes applied were to help the targeted community effectively.

The project is aimed at developing a web system enabling immigrants and refugees in the UK to empower themselves to seek and search for help by effective time margins. The project was initially focused on providing a database of social services departments in the UK depending on the geographic location of a person, with a translation module, enabling the end-user to translate the information into his/her native language. However, as the project continued and field research started to cultivate, it was evident that the initial plan for the development of the system would solve only half the problem faced by our targeted audience. The project blueprint evolved into a system that would be capable of connecting an end-user to a relevant counsellor, empowering the user to get a quick and accurate response to their queries.

The evolution of the project from a simple database system to a proper query and connect chat system was greatly influenced by the survey conducted to filter out the core issues of the community and their most effective software solution. The project experienced maturity with facts coming from the actual audience and hence the complexity of the project greatly increased. The core functionalities were sorted out to be:

- Data Upload
- Content Translation
- Query Management
- Chatting Module (Text/ Audio/ Video)

These main functionalities however had to be generalized and some initially sorted out features were ignored in the final implementation of the system due to time and development resources constraints. The system after development went through a proper testing life cycle and each module was tested against several test cases. The applied test cases are included in this project and it should be noted that the project was tested using automation testing techniques along with the manual testing routine.

1.2 Project Aims and Objectives

The project went through an evolutionary cycle as explained in detail previously. The initial aims and objective were changed, and the most important ones were selected to be the baseline of the system for development. These shortlisted objectives encompassed all the major requirements of the system as well as provide a baseline for any new upgrade that may be achieved in the future. The shortlisted objectives are as follows:

Objective 1 – Research and analyse the current migrant and refugee situation in the UK, to determine if the project is needed.

Objective 2 – Brainstorm questions for the development of a survey to find out the actual issues faced by the targeted community in their daily life.

Objective 3 – Survey by both online and offline means and hide the identity of the people who submitted the survey.

Objective 4 – Analyse the results of the survey:

- To filter out the core issues.
- To understand the social issues faced by the community.
- To identify technical sources for mitigating the identified issues.
- To develop some analytics from the survey for future work.

Objective 5 – Develop a paper prototype of the system and undergo brainstorming sessions to finalise all the functional requirements.

Objective 6 – Implement the prototypes into a web-based system and design test cases against each functionality to assess its usefulness and iteratively fixing the identified bugs.

Objective 7 – Deploy the solution in a beta environment.

1.3 Approach

This project was split into different phases and sub-phases, a divide and conquer approach was adopted to achieve maximum productivity. The main phases of the project can be divided into two distinct ones. The first one was to develop a keen understanding of our audience and the subject under study. As explained previously, a significant amount of time and resources were utilised to build a root level understanding of our requirements and our targeted communities. After proper field research of the existing processes for aiding the migrant community, a set of core goals and objectives were finalised for the final implementation. The first phase marked the end of all brainstorming and research for the core features and problem identification. Phase 1 included interviews with the target audience and required ethical approval. The necessary ethical approval documents can be found in Appendix A.

The second phase of the project consisted majorly of the implementation and development of the system, based on the requirements and processes identified in the first phase. This phase involved mostly the development of paper prototypes and then technical decisions. The selection of tools and technologies used were focused on the available skillset and the efficiency of the individual technology. It should be noted that all the respective methodologies of software engineering were applied and, in the essence, this system closely resembles a CRM system as well. Keeping that in view, a blend of the waterfall model with development specific scrum techniques were used for the software development lifecycle.

1.4 Technology stack

The project development was dependent on multiple factors and many diverse technologies and set of tools were available. However, after proper experimentation and research, the following final stack was used for the entire process.

1.4.1 Backend Technology

There are several frameworks available for backend development. The criteria for the project was decided to be focused on the scaling of the system. The nature of the project is that it can be scaled towards more use cases and complexity. The selected framework for this project was finalised to be **“Python Django Web Development Framework”**. The comparison between the two famous frameworks of python, i.e “Django vs Flask” is as follows:

Django	Flask
This is a full-stack web framework built on the principles of the batteries-included approach.	This is a lightweight framework that offers minimalistic features.
Django features many predeveloped features, which are highly stable and can be accessed by the developers properly in all needed scenarios.	Flash is a minimalistic framework and does not in general contain powerful predeveloped features.
Django features a built-in “Django Admin” interface, and this is one of the most powerful admin interfaces available with any other framework, this can utilize by the system owner in a very intuitive manner.	Flask does not support or contain any administrative package at all, the 3 rd party available packages are also not powerful enough to match the effectiveness of Django’s built-in Admin interface.
It comes with a built-in template engine that is used to make the front end of the applications.	Flask does not have a native template engine, rather it uses jinja 2 (Django’s template engine)
Django features the concepts of apps inside a project, it allows the developer to segment the project into different applications, this simply makes it much more maintainable and compact, it also reduces the chances of conflict, which going through multi-branch code configuration.	Each project is treated as a single application, developers may use some custom file division approach but in general, Flask does not offer application segmentation inside a project.
The bootstrapping tool helps developers build scalable web designs inside the application.	The Flask admin builds in features is not as prominent as in Django.
Django features a very powerful Database ORM, which is compatible with most of the available SQL databases and there is virtually no need to write raw queries anymore, the ORM provides support for all major fetching and insertion methods.	Flash does not feature a proper stable ORM, developers have to write raw queries to solve or perform common tasks, these queries may also need to be manually made compatible with different SQL versions.
Django is a full-fledge production-ready framework, which can perform multi-threading as well, it is also capable of handling multiple simultaneous	Flash is considered to be single-threaded and is not suitable for large scale applications

Django offers many out of the box solutions and is considered to be a faster development tool when compared to others, it is also popular to contain different sub tools which make it more intuitive.	Flask is a good start if you are getting into web development. There are many websites built on the flask and gain heavy traffic, but not as much compared to the ones in Django.
Django is a better choice for a project which do not change the requirements often, since Django is targeted for large scale application development, it also has constraints that make it tough to accommodate changes in certain situations.	Flask is usually used to create simplistic applications and functionalities can be changed with little effort, but the trade-off is its capacity of constructing smaller and simpler applications.
Developers are not encouraged to make changes in the core packages provided build-in Django itself.	Developers are free to use any plugins and libraries and flexibly build functionalities, since the mandate of Flask is simple and saleable application development, the usage of the plug-in is not an issue at all.
Django is suited for more significant projects which focus on power functionalities and use more processes resources, its core features enable it to utilize the available computational resources in such a way that the system does not overload with requests.	Flask is simple and does not enforce any kind of specific pattern to develop the functionalities and is considered very good for small application, which does not need more computational resources than the initially allocated ones.
Django framework ensures developers use the best design patterns and the proper template architecture style, this is because proper documentation and efficient system design is the root and main idea behind Django.	Flask is very much open-ended, and it does not enforce the use of any specific design patterns and it also does not utilize any specific template architecture.
The only big disadvantage of using proper template architectures and designs patterns is the existence of more Lines of Code in the development process, it is estimated that Django takes twice as many lines of code for functionality as a flask, but on the other hand, the code generated in Django is more efficient and stable.	Flask application utilities fewer lines of code of a given functionality but it may not be as stable as the same functionality developed in Django using the enforced design patterns.

Table 1 Django versus Flask

1.4.2 Frontend Technology

The frontend of the application was made using **HTML, CSS & JavaScript** and it was later encapsulated inside the “Django Templates” and message sending was made possible using the

template tags or Jinja 2 language. It should be noted that the front end is coupled in the same single project as is the backend.

1.4.3 Storage

The project database is moderated and is balanced between insertion and fetching and for this reason, the SQL database was selected. The project uses SQL lite database which comes preconfigured with every Django project. It is easy to use and the insertion and fetching delays are ideal and do not cause delays.

1.4.4 Code Environment

The project was contained inside a virtual environment using the anaconda virtual environment system made for python. The usage of the environment makes the code more protected and the development packages remain unaffected by any external changes.

1.4.5 Project Management

The project management was managed by using “Trello Boards” and the board signified the types of tasks and the sub tasks for a week in each of the category. The project consisted of a single member-only, but the board helped in keeping track of the activities and served as a to-do list as well.

1.4.6 Software Testing

The project testing was given much importance and a full fledged test plan was made with multiple test cases for each module. The testing was limited to core functionality due to the constraints of time and resources. The project was tested using manual testing techniques and some regression-based modules were also tested using the automation techniques but utilizing the popular testing framework “Cypress.io”.

1.4.7 Google Tools

The translation module of the system was developed using the google translation API and the survey for the project was also conducted using “Google Forms”. The interactive environment provided by google forms suits the survey needs at best and it is also very useful for generating key infographics from the survey data.

1.4.8 Code Configuration

The code configuration was done using a private repository on GitHub, it is free to use and very intuitive. The project was pushed on the repository after every significant change and added features. This helps in maintaining a proper track record of what change occurred at what time and how it affected the system. The use of GitHub for projects is highly encouraged and this project became very maintainable through the proper utilization of code configuration.

2 Background

The previous and current century has seen several devastating wars and natural atrocities, which caused migration from one area into another, in some cases the migrations remain internal and in other cases, the migration stretched from one country to another. This started newer issues for the migrant community. It is very hard to leave one's own country and move to another one and start anew with everything. There are various issues faced by these communities, but the most prevalent ones were related to cultural and language gaps. The UK is home to many different people belonging to different cultures and many of these community members have come to the UK seeking better employment opportunities and permanently settle here as contributing member of the society. The UK's immigration system was already stretched to meet the demands of the rising cases of immigration.

Everything started to become much harder as the pandemic started, the existing applications went on hold and the offices closed due to lockdowns. These communities were the most affected during the pandemic. The reason for this was mostly because of their language gaps and their inability to utilize the available resources due to many different reasons. These communities were lacking in adequate legal help as well as a medical help. The reasons can be argued upon to be very diverse, but we have concluded that these issues are mainly due to the information and effective communication gap. It must be kept in mind that there is a very clear difference between communication and effective communication. Effective communication is a communication in which the whole subject and details of a discussion are conveyed between the sender and the receiver. This exchange of information in an effective manner becomes hard between people belonging to different cultural backgrounds.

There are multiple differences between the norms of people coming from different areas of the world and the natives of the UK. The people serving in the state departments tasked to deal with the migrant community are often hampered with the language barriers and hiring interpreters for every specific language is not cost-effective at all and with the world turning into a global village it is no longer humanely possible to cater the need to bypass the language barrier. The UK has successfully incorporated technology in various departments and has digitalized itself to a very effective extent. The healthcare sector has greatly benefited from the adoption of technology and software-based automated solutions.

The refugee facilitation processes have also been technically transformed but there is still room for more technical enhancement. It should be understood that not every issue or problem that a common migrant face is related to the state but rather often these issues are related to basic information. This basic information needs to be available in a very simplistic way and it needs to be easily accessible to everyone who wants to use it.

2.1 Survey

To fully understand the issues and needs of our targeted community. A dedicated survey was devised after many different sessions of brainstorming with a volunteering community member. The set of 13 questions finalized for the survey are as follows:

1. What were the main problems you faced once you arrived?
2. Do you face issues with communication due to the language barrier?

3. What is your English language proficiency?
4. How many years you have been in the UK?
5. What was the major problem faced in resolving the communication/language barrier?
6. What were the issues faced in finding work/job?
7. Have you encountered issues in claiming social services?
8. What major problems did you face while claiming your social rights?
9. How much does the pandemic affect you?
10. Rate the difficulty in finding the right transportation services?
11. Rate the difficulty in finding the nearby GP (General Physician / Doctor)?
12. How successful you have been in communicating your problems to the authorities?
13. Do you think web-based services would help you resolve your issues?

This survey was conducted using chiefly “Google Forms” however some physical responses were also conducted as well. The identity of the participants has been kept anonymous due to data privacy policies and the consent of the participants as well. The survey itself gave a very much needed statistical support to the project’s initial hypothesis. The survey highlighted the much-needed insights regarding the issues for the targeted community.

2.1.1 Survey Results:

The survey was filled by 57 participants belonging to the targeted community for this project. The main highlights are as follows:

1. Most of the Refugees settled in the UK / foreign countries are facing issues due to communication (63 %). The language barrier is a serious problem for migrants in the UK.
2. 40% of respondents say that they have average (3 on a scale of 1-5) proficiency in English. This has a knock-on effect on other issues faced by refugees e.g., access to transportation and GP.
3. 38.6% of respondents conveyed that the pandemic has affected them and they are facing problems due to ongoing COVID situations.
4. The majority of the participants are in favour of resolving their issues with the help of some web-based services (72 %).
5. A massive 61.4 % per cent of the people voted “communication problem” as the main issue for claiming their social rights.
6. The majority of the participants highlighted that in only 20% of the times, they have successfully conveyed their problems to the authorities.

After the insights obtained through the survey, the hypothesis of this project became a very significant reality, and the project contains field information strengthening the need for the solution. The number of participants was limited, but this was due to low outreach to the targeted community as the project was being implemented under a strict lockdown and with a limited number of resources.

2.1.2 Survey Infographics

The graphs constructed from the survey are as follows:

What is your English language proficiency?

57 responses

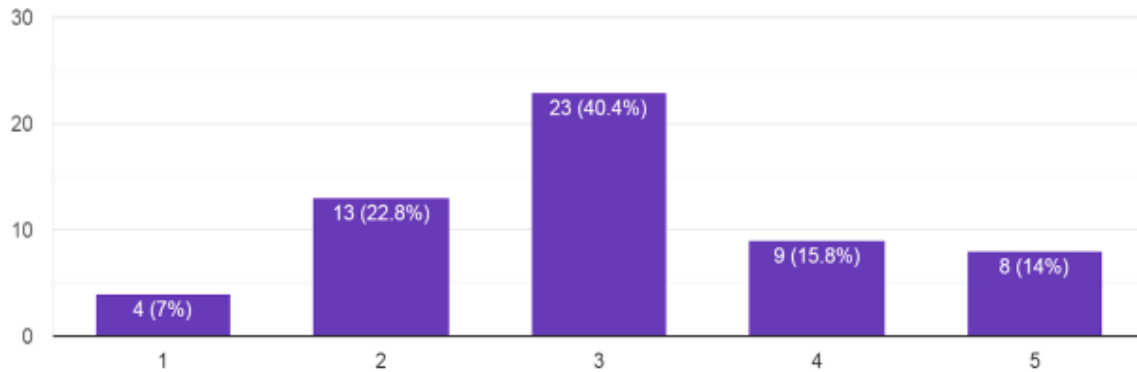


Figure 1 Histogram Showing English Proficiency

Participants were asked about their English Language proficiency to enable a greater understanding of their needs regarding what language would be suitable for the application and to understand some of the language difficulties faced. Participants were asked to rate their English language proficiency from 1 to 5. With 1 being little understanding and 5 being fluent. The results of this question can be found in figure 1.

How many years since you have been in UK?

57 responses

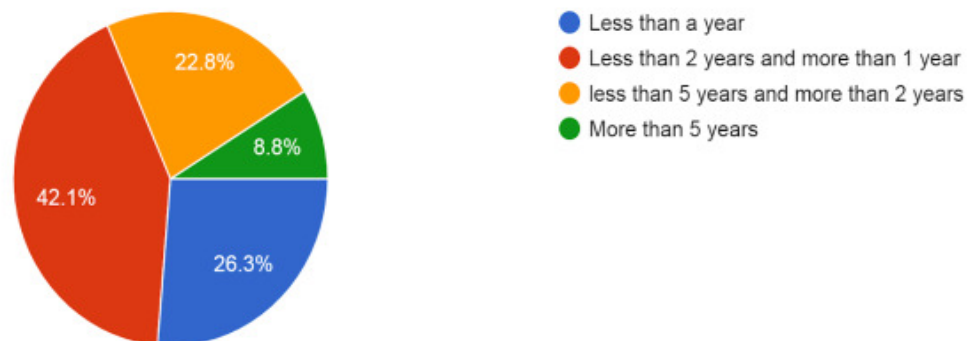


Figure 2 Histogram Showing migrants duration in the UK.

The figure above illustrates the period each participant spent within the country so that the difficulties of newcomers could be evaluated. For example, the survey showed that people being in the country for a long time have found that the country policy is welcoming and getting used to the environment is achievable. People who have just come to the UK are not feeling as secure as those being here for a longer time due to the language barriers and the challenges they are facing. This question was asked to establish if people are facing the same issues and if staying longer in the country resolves the initial issues that people faced. Depending on the results, refugees having overcome the difficulties at the beginning can provide support and advice to those who have just come and guide them through the journey.

Have you encountered issues in claiming social services?

57 responses

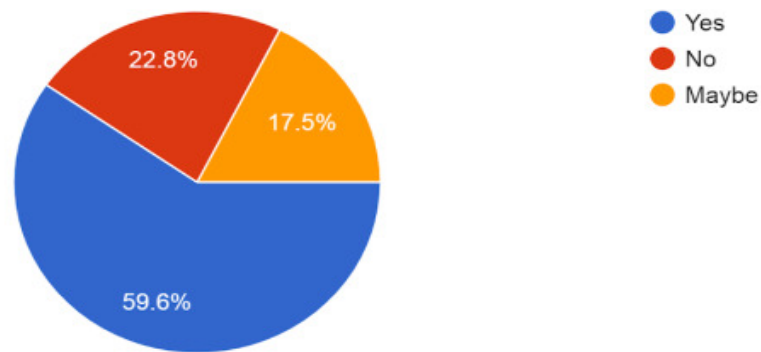


Figure 3 Histogram Showing migrants satisfaction rate of the social services.

The figure above shows the percentage of people satisfied with social services. It became obvious that more than half of the participants faced issues when reaching out to social services, e.g., claiming benefits.

Rate the difficulty in finding the right transportation services:

57 responses

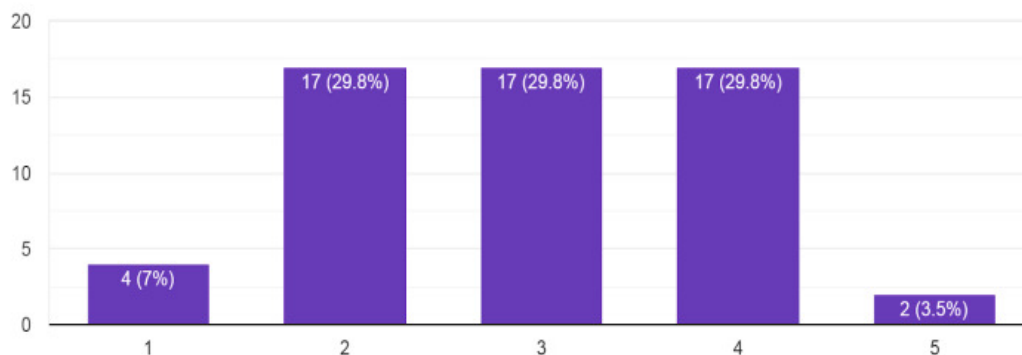


Figure 4 Histogram Showing rate of migrant difficulty in finding transport

Participants were asked if they find it difficult in finding the right transportation service and rate the difficulty from 1 to 5. With 1 being a little difficult and 5 being extremely difficult. The majority expressed the opinion that transportation is not that challenging.

What was the major problem faced in resolving communication / language barrier?
57 responses

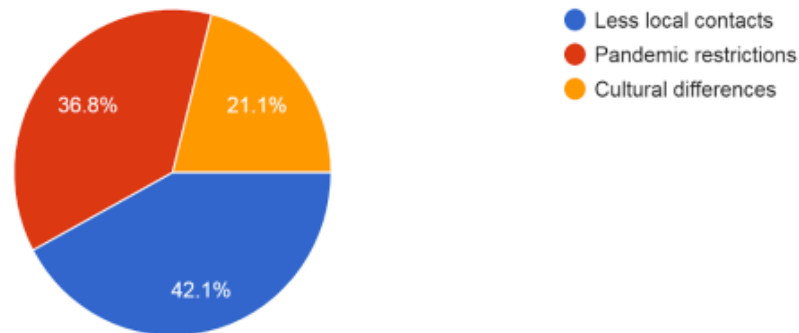


Figure 5 Histogram Showing rate of the reasons for language barrier

The figure above shows the problem related to the communication and language barrier. As the main issue discussed in this report, the survey also proved that communication is not as efficient as it should be. Quite a large percentage of the refugees have a constraint circle of local contacts. This issue has been exacerbated due to the pandemic situation and the negative impact it has had on this community in the country. In addition, the cultural differences make it difficult for refugees to interact, socialise and get familiar with the environment and the local culture and habits.

Rate the difficulty in finding the nearby GP (General Physician / Doctor) :

57 responses

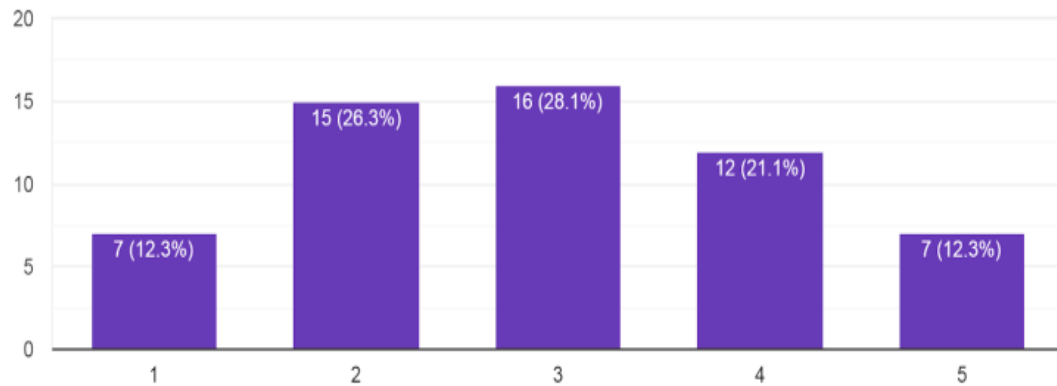


Figure 6 Histogram Showing migrant's difficulty rate on finding GP.

Having access to health services is one of the most important aspects for newcomers. The graph above rates the difficulty from 1 to 5 with 1 being least difficult and 5 extremely difficult. Most participants rated the difficulty as average with access being provided when needed.

How successful you have been in communicating your problems to the authorities?

56 responses

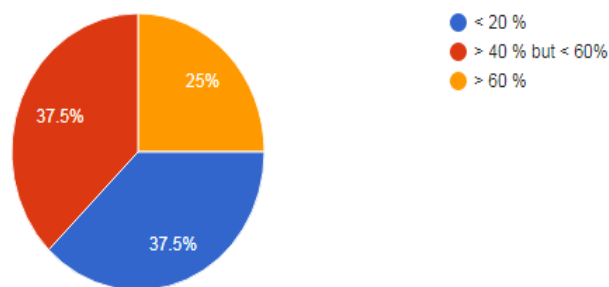


Figure 7 Histogram showing the rate of migrant's success in communicating with authorities.

Getting in touch with the right authority is essential. The survey results show that most of the people were struggling to find the relevant authority for different aspects. Only 25% of the participants were successful in resolving their issues.

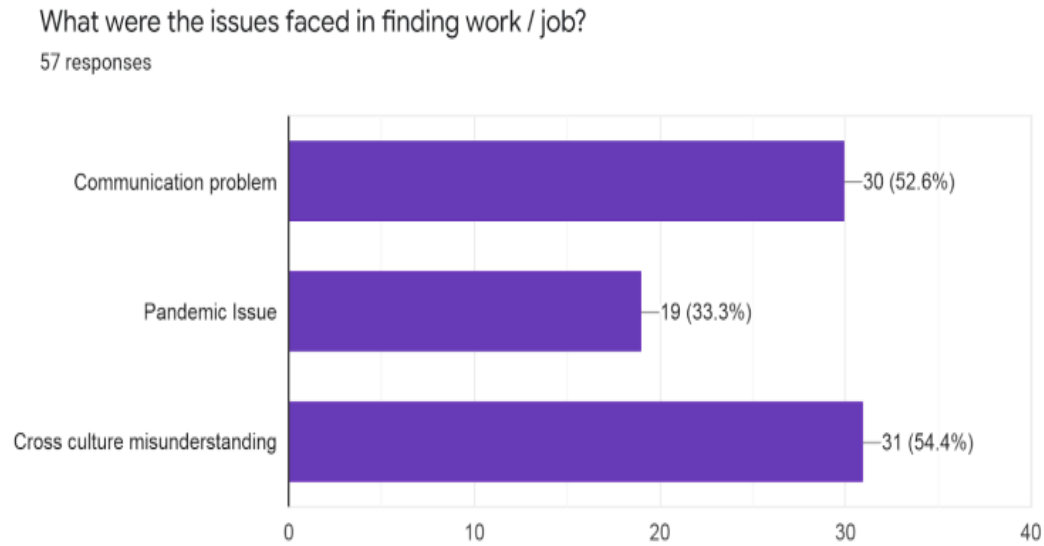


Figure 8 Histogram Showing issues faced in finding work/Job.

In this report, it has been discussed that refugees come to the country to seek a better quality of life by finding better job opportunities, getting a degree, having better social and health and care services. Thus, finding a job is of great significance. The main reasons for facing issues when looking for a job was found to be cross-culture misunderstanding and the communication problems which arises from the lack of basic language knowledge.

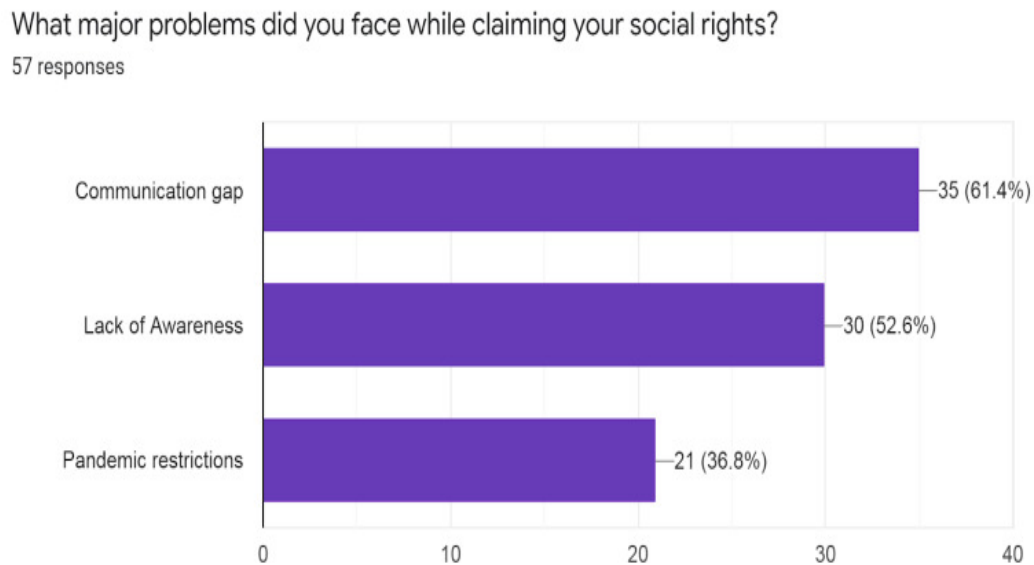


Figure 9 Histogram Showing kind of problems migrants face when claiming social rights.

The same trend was observed in the issues related to claiming social rights. The main issue was again related to the communication gap between the refugees and the social services. The idea of this figure above was to establish if the problems are arising due to the refugees not being able to communicate across their needs and situation or the social services were not able to provide the correct solution.

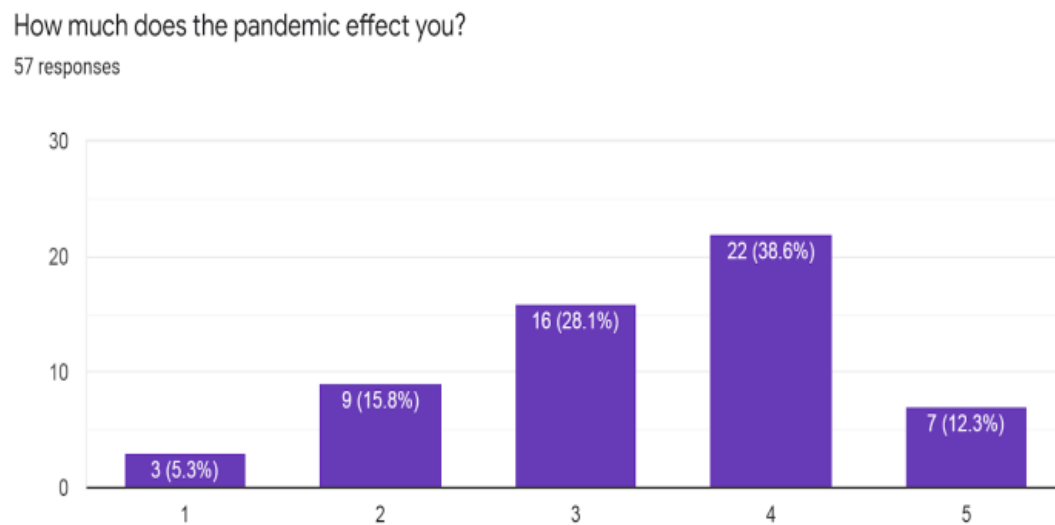


Figure 10 Histogram Showing the degree Covid19 affects migrants.

During the development of this report, the worldwide pandemic affected the whole population to a great extent. Refugees were asked to rate the effect of the pandemic from 1 to 5 with 1 being a minimal effect and 5 being a huge effect. Most of the participants rated the pandemic impact with 4 which means that most of them have been influenced significantly by not being able to find jobs, difficulty in contacting authorities and experiencing mental health issues due to lockdown and lack of socialising.

Do you think web based services would help you resolve your issues?

57 responses

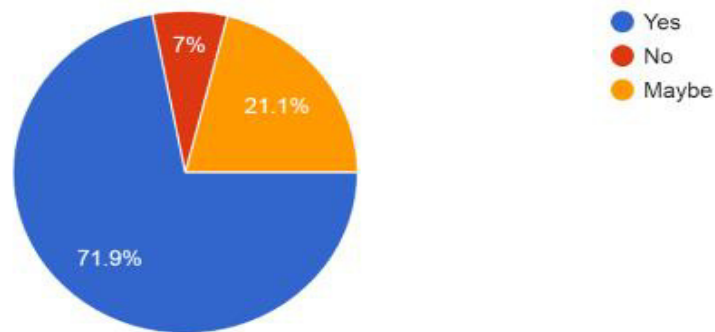


Figure 11 Histogram Showing the percentage of immigrant's viewpoint if web based services solves their issue

The last survey was related to the web-based services provided. Most of the participants are happy with the web-based services and they believe that issues could easily be resolved through online communication because of the language options it provides. Some people prefer to have face to face opportunity to access services as it makes communication more efficient. Only 7% voted as web-based services being not efficient some of the refugees do not have access to online systems and the technology ignorance.

Each of these graphs represents the participant responses in a very meaningful manner. These graphs are easy to understand and are all generated automatically using google forms.

2.2 Existing Systems

This project deals with a very specific problem and there are not many solutions like this available for usage in the UK. The existing systems had to be searched to analyze them. This also highlights the lack of awareness among the people regarding solutions dealing with the refugee and migrant problems. A study of the existing solution is always very important because it highlights the pros and the cons of the systems which are in use. The pros and cons identified in each system helps in refining the solution being developed. The details regarding the identified systems are as discussed below.

2.2.1 Refugee Cardiff

This is a website focused on the refugee community in Cardiff. It provides details regarding the following key areas (Refugee Cardiff, 2021):

- Child and Family support
- Refugee information & support
- Seeker information & support
- English classes
- Food
- Clothing

- Women
- Sports
- Volunteering
- Legal Advice

These are very important areas for the refugee community in Cardiff but there are many areas that this website is lacking. The first thing which is noticed for any digital system is the 'ease of use' for the user. One website may have very good functionality in it but it may be useless if the end-users do not find it attractive. The main components of the User experience focus on the usability of the system and its attractive design. In the past, there was not much progress in the designs of the digital systems but now, there is much focus on the look and feel of the system. It is now understood that the easier a system is to use, the more useful it will become.

The design of the website is very much cluttered, and everything seems to be unclassified, the side menu of the website is not very stable and on a few screens sizes the menu appears to be cut off from one side.



Figure 12 Refugee Cardiff Menu

It is very much visible that the 4th and 5th item in the menu panel is overflowing to the left side, the menu bar itself seems to be providing details in an unordered manner. The various activities need to group by some criteria. The overall design of the website is adequate, but it is certainly not ideal as much of the text colour is closely resembling the background colour, which makes it very a nuisance to be read properly for people having eyesight problems.

The other problem concerns the language barrier. The website assumes that all the users are proficient with the English language. While that is not the case at all, it has been discussed in detail in this report that there is a vast majority of refugees who are not English literate and need to be able to read things in their native languages. The survey conducted also highlights a 63% sample

population has problems understanding details in the English language and would prefer using their native language to read important details. There is no option available on the website which enables the users to select a language and translate the content into their preferred language.

There is also no system query registration system on the website. A user cannot book an appointment with a professional to guide them towards the solution of their problems. There is also no chatting or calling feature on the website. The website assumes that the provided textual information is enough for the end-users to solve their problems. The case in reality is quite different from the assumptions made. There is a very crucial need for a live call feature between a counsellor willing to provide help and a refugee or migrant who needs genuine help. The pros and cons matrix is made from this website.

Pros	Cons
<ol style="list-style-type: none"> 1. Provides help for different areas. 2. Provides focus on the legal issues. 3. Provides to the point details of the concerned topic. 	<ol style="list-style-type: none"> 1. The User Experience of the website is not good. 2. There is no translation feature available. 3. There is no query or request management system. 4. There is no calling or live chat system.

Table 2 pros and cons of the Refugee Cardiff website

2.2.2 Welsh Refugee Council

This website is focused on the refugee community in Wales. The main aim of the website is to help asylum seekers and refugees get registered and gain rights within the country (Welsh Refugee Council, 2021). The 'Welsh Refugee Council' is not an online service, it is rather a non-government organization providing facilitation to the refugee community. This website does not focus on other local or social services but is a platform to provide legal registration facilitation for refugees coming into Wales.

The first feature that catches the eye of the user is its design, the design of this website is very attractive, easy to use, and the information placed on it is not cluttered together at all. The overall flow of the website is also very good and does not confuse the user too much regarding what to do. The second thing on the website is that it provides the translation in 'Welsh', although this may be a step forward for effective communication, it seems not to be very useful, it must be kept in mind that the refugees are not natives of Wales or the UK, they are coming from different cultures with different languages, and they are probably not equipped with the local languages of the Host country. The website however does provide some static translations for specifically selected languages for some menus, which is a very good approach as here many refugees speaking the provided languages would be able to understand the content.

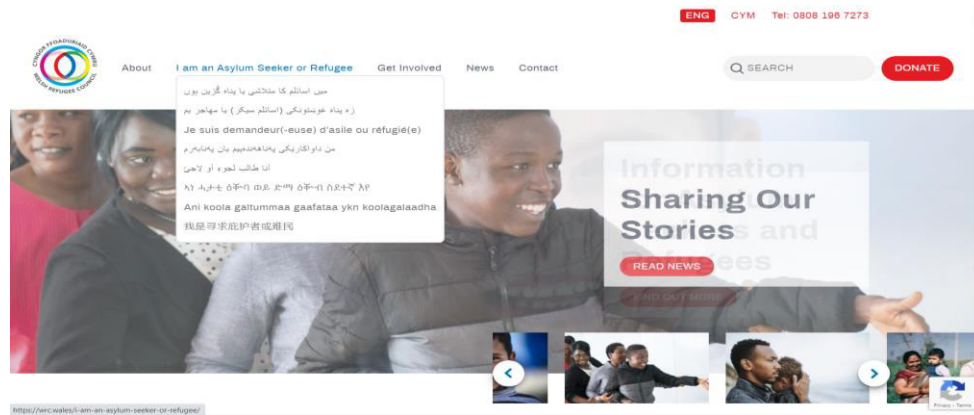


Figure 13 Welsh Refugee Council

The website also features a banner which states a toll-free number for refugees to call upon when they need assistance, the instructions about making the calls are published in individual banners translated in 8 different languages. The issue here is that telephone calls do not maintain a certain history of the past conversations, the actual issues and cannot maintain a proper record of issues discussed. It is a very good initiative and would be very helpful for a great many people, but it would not be very efficient in the longer prospect of the system. The pros and cons matrix of this website is pasted below.

Pros	Cons
<ol style="list-style-type: none"> 1. Good User Experience 2. Good Design 3. Some translations Available 4. Toll-free calling 	<ol style="list-style-type: none"> 1. Does not focus on any local services. 2. Does not provide more translation for the entire content. 3. Does not have an in-built chat feature

Table 3 Pros and Cons of the Welsh Refugee Council website

2.2.3 Sanctuary (Welsh Government)

This website is maintained by the Welsh government and provides details on various areas for the refugees and the migrant community (Welsh Government, 2021). The information available on this website is the most accurate one available on any online platform since the information comes directly from the government. This website cannot be fully compared with the other systems that have been discussed previously. This website has many good features, and the most prominent ones are:

- Good User Experience & Design
- Contains information on diverse topics.
- Features translation for content in many different languages
- Features a text recitation module as well, which can recite the text in the selected language.

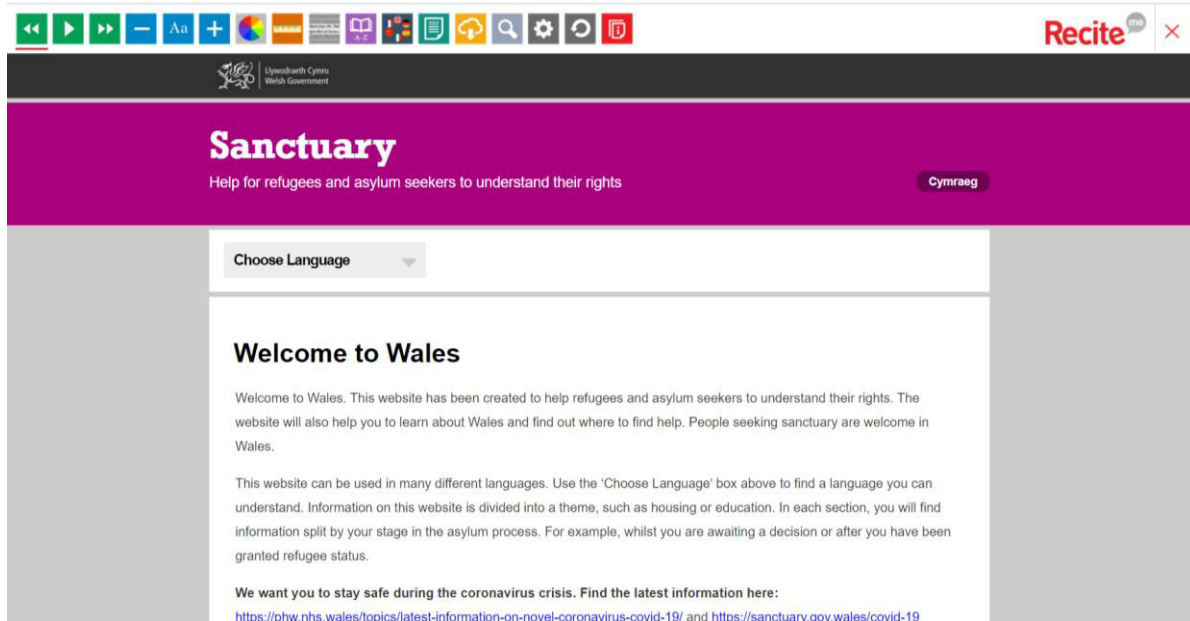


Figure 14 Sanctuary Home Page

The features of this website are very good, but there is a certain area that this website lacks and that is the availability of live communication with professionals which can provide help and the lack of a proper dashboard, where the user can keep track of his/her requests and their responses. These things are the key attributes missing from this website and hence making it a very good but incomplete solution to the problem at hand, which is helping the migrant and refugee community most efficiently and easily possible. The pros and cons matrix of this system is pasted below:

Pros	Cons
<ol style="list-style-type: none"> 1. Good User Experience 2. Good Design 3. Translations and recitation Available 4. Accurate information 	<ol style="list-style-type: none"> 1. Does not have an in-built chat or call feature. 2. Misses the proper request management dashboard functionality.

Table 4 Pros and Cons of the Sanctuary (Welsh Government) website

This concludes our study of the existing system and through the analyses of these websites and with the survey information at hand, it is evident that the features focused on this project are very important and are not currently developed or unified on a single platform, making the rationale of this project more concrete than the initial hypothesis.

3 Approach

3.1 Functional Requirements

Functional Requirements describe the expected functionality-based attributes of the system. They define the functionality details regarding the system (Alexsoft, 2018). This project has been different types of required functionalities and these have been identified iteratively throughout the life of the project. The

functional requirements have been written in a specific format, which divides our broader functionality into sub-requirements, hence representing the entire feature into a stepwise list which makes it very easy for the developer to implement it, there is an acceptance criterion written with every functional requirement. As explained previously the software lifecycle model chosen for this project is a hybrid of water fall and scrum, so the technical documentation style is also a blend of these two engineering model paradigms.

The following Functional requirements are identified for the system.

FR01: Super Admin - log in (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR01-01	The system shall interface the Super Admin Panel to the Super Admin	The Super Admin is Successfully logged in the system
FR01-02	The system shall enable the super Admin to insert his/her credential in the username and password fields	
FR01-03	The system shall verify the credentials provided by the user and perform authentication.	
FR01-04	The system shall navigate the super admin to the super admin panel successfully.	
FR01-05	The system shall not navigate to the super admin panel in case of wrong credentials	

FR02: Super Admin - View All (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR02-01	The system shall enable the Super Admin to view all database tables in the interface	The Super Admin is Successfully able to view all the database models in the system.
FR02-02	The system shall enable the Super Admin to view sub-records of all database models in the panel	

FR03: Super Admin - Create Users (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR03-01	The system shall enable the Super Admin to access the user's model.	The Super Admin is Successfully able to create new users in the system.
FR03-02	The system shall enable the Super Admin to create new users.	
FR03-03	The system shall enable the Super Admin to assign new users different roles.	

FR04: Super Admin - Create Roles (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR04-01	The system shall enable the Super User to access the Roles in the panel.	The Super Admin is Successfully able to create new roles in the system.
FR04-02	The system shall enable the Super User to create new Roles with a given set of permission sets.	

FR05: Super Admin – Set Password Email (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR05-01	The system shall automatically send a set password email to the newly created Super Admin/user.	The System should send an automated set password email to the newly created user and save the given password.
FR05-02	The system shall not allow a user to log in without setting up a password.	
FR05-03	The system shall enable the new user to open the emailed link and present the set password page.	
FR05-04	The system shall enable the new user to insert his/her new password in the password field.	
FR05-05	The system shall save the user's password and allow the user to log in to the system using the newly created password.	

FR06: Super Admin – Forget Password (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR06-01	The system shall provide an option for 'Forget Password' on the login page.	The Super Admin is Successfully able to reset his/her password in the system.
FR06-02	The system shall present the reset password screen.	
FR06-03	The system shall enable the super admin to insert his/her email address.	
FR06-04	The system shall email the reset password link to the super admin email address.	
FR06-05	The system shall enable the user to access the reset password → set password screen to set up a new password.	

FR07: Super Admin/Admin – Insert Local Services (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR07-01	The system shall enable the admin to insert data in fields for adding a new local service	The Super Admin is Successfully able to insert local services into the system.
FR07-02	The system shall save the new local service added by the user.	
FR07-03	The system shall update the list of local services everywhere in the system.	

FR08: Super Admin/Admin – Insert Departments in Local Services (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR08-01	The system shall enable the admin to insert a new department inside a local service created	The Super Admin is Successfully able to insert departments in local services in the system.
FR08-02	The system shall save the new department in the selected local service.	
FR08-03	The system shall update the list of all departments inside local services everywhere in the system.	

FR09: Super Admin/Admin – Insert Social Services (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR09-01	The system shall enable the user to insert data in fields for adding a new social service	The Super Admin is Successfully able to insert social services into the system.
FR09-02	The system shall save the new social service added by the user.	
FR09-03	The system shall update the list of social services everywhere in the system.	

FR10: Super Admin/Admin – Insert department in Social Services (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR10-01	The system shall enable the admin to insert a new department inside a social service created	The Super Admin is Successfully able to insert departments in social services in the system
FR10-02	The system shall save the new department in the selected social service.	
FR10-03	The system shall update the list of all departments inside social services everywhere in the system.	

FR11: Super Admin/Admin – Edit Local Services (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR11-01	The system shall enable the admin to edit data in fields for adding a new local service	The Super Admin is Successfully able to update details of local service in the system
FR11-02	The system shall save the edited local service added by the user.	
FR11-03	The system shall update the list of local services everywhere in the system.	

FR12: Super Admin/Admin – Edit Departments in Local Services (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR12-01	The system shall enable the admin to edit a new department inside a local service created	

FR12-02	The system shall save the edited department in the selected local service.	The Super Admin is Successfully able to update details of any department of a local service in the system
FR12-03	The system shall update the list of all departments inside local services everywhere in the system.	

FR13: Super Admin/Admin – Edit Social Services (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR13-01	The system shall enable the user to edit data in fields for adding a new social service	The Super Admin is Successfully able to update details of local service in the system
FR13-02	The system shall save the edited social service added by the admin.	
FR13-03	The system shall update the list of social services everywhere in the system.	

FR14: Super Admin/Admin – Edit department in Social Services (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR14-01	The system shall enable the admin to edit a new department inside a social service created	The Super Admin is Successfully able to update details of any department in social services in the system
FR14-02	The system shall save the edited department in the selected social service.	
FR14-03	The system shall update the list of all departments inside social services everywhere in the system.	

FR15: Super Admin/Admin – Delete Local Services (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR15-01	The system shall enable the admin to delete a local service.	The Super Admin is Successfully able to remove a local service from the system
FR15-02	The system shall delete the local service along with all linked departments.	

FR16: Super Admin/Admin – Delete Departments in Local Services (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
----------	-------------------------	---------------------

FR16-01	The system shall enable the admin to delete a department inside a local service.	The Super Admin is Successfully able to remove a department from a local service in the system
---------	--	--

FR17: Super Admin/Admin – Delete Social Services (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR17-01	The system shall enable the user to delete a social service.	The Super Admin is Successfully able to remove a social service from the system
FR17-02	The system shall delete all the linked department in the social service as well.	

FR18: Super Admin/Admin – Delete department in Social Services (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR18-01	The system shall enable the admin to delete a department inside a social service.	The Super Admin is Successfully able to remove a department from a social service in the system

FR19: Super Admin - Edit Users (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR19-01	The system shall enable the admin to insert new details in the user fields.	The Super Admin is Successfully able to update user details in the system
FR19-02	The system shall enable the admin to save changes made in the user data.	

FR20: Super Admin - Delete Users (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR20-01	The system shall enable the admin to delete a user.	The Super Admin is Successfully able to delete a user in the system

FR21: Super Admin - Edit Roles (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR21-01	The system shall enable the admin to give or take permissions in a role.	The Super Admin is Successfully able to update role details in the system
FR21-02	The system shall enable the admin to change the name of the role.	
FR21-03	The system shall save the changes made by the Admin/Super Admin	

FR22: Super Admin - Delete Roles (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR22-01	The system shall enable the admin to delete a role.	The Super Admin is Successfully able to delete a role in the system
FR22-02	The system shall not delete assigned users in the deleted role.	
FR22-03	The system shall assign the assigned users in the deleted role to a default role.	

FR23: Admin – View User Queries (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR23-01	The system shall enable the admin to view the latest requests for meetings or queries made by the end-users (site users → refugees)	The Admin is successfully able to view all user queries in the system

FR24: Admin – Assign Counsellor to User (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR24-01	The system shall enable the admin to view a list of available counsellors in the relevant department.	The Admin is successfully able to assign a counsellor to user queries in the system
FR24-02	The system shall enable the admin to assign a counsellor a user request.	
FR24-03	The system shall notify the user via email that a request has been assigned.	
FR24-04	The system shall notify the counsellor via email that a new request has been added to a queue.	

FR25: Admin – Dismiss a User Request (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR25-01	The system shall enable the admin to either accept or dismiss a user query.	The Admin is successfully able to dismiss/ reject a user query in the system
FR25-02	The system shall enable the admin to assign a counsellor a user request.	
FR25-03	The system shall notify the user via email that the request has been dismissed.	

FR26: User – Sign up (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR26-01	The system shall enable the user to access the sign-up page.	The end-user is successfully able to sign up in the system
FR26-02	The system shall enable the user to fill in all the requirements.	
FR26-03	The system shall register the new user in the system.	

FR27: User – Sign in (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR27-01	The system shall enable the user to access the sign-in page.	The end-user is successfully able to sign in to the system
FR27-02	The system shall enable the user to enter his/her username and password.	
FR27-03	The system shall authenticate the provided user credentials.	
FR27-04	The system shall sign in the user in case of correct authentication.	

FR28: User – Language Selection (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR28-01	The system shall enable the user to view the list of available languages.	The end-user is successfully able to change content the system
FR28-02	The system shall enable the user to select his/her preferred language.	

FR28-03	The system shall change the website language depending upon the selected language.	
---------	--	--

FR29: User – View Local Services/Departments (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR29-01	The system shall enable the user to view the list and details of available local services and the department.	The end-user is successful to view available services the system
FR29-02	The system shall change the website language depending upon the selected language at the start of the user session.	

FR30: User – View Social Services/Departments (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR30-01	The system shall enable the user to view the list and details of available social services and the department.	The end-user is successful to view available services the system
FR30-02	The system shall change the website language depending upon the selected language at the start of the user session.	

FR31: User – Create New Query/Requests (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR31-01	The system shall enable the user to select the category of his/her complaint/query.	The end-user is successfully able to create new queries in services the system
FR31-02	The system enables the user to insert details about the issues.	
FR31-03	The system saves the query and sends it for admin approval.	

FR32: User – Request Meeting Directly (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR32-01	The system shall enable the user to select the category of his/her complaint/query.	The end-user is successfully able to request meeting directly in the system
FR32-02	The system enables the user to request a direct meetup with a counsellor.	
FR32-03	The system saves the query and sends it for admin approval.	

FR33: User – View all requests (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR33-01	The system shall enable the user to view all active requests and their status.	The end-user is successfully able to view all self-generated requests in the system

FR34: User – Filter all Requests (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR34-01	The system shall enable the user to apply defined filters on the request page.	The end-user is successfully able to search/ filter all self-generated requests in the system
FR34-01	The system shall display the filtered requests.	

FR35: User – Edit Query (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR35-01	The system shall enable the user to edit certain fields of his/her requests.	The end-user is successfully able to update existing queries in the system
FR35-02	The system updates the changes made by the user.	

FR36: User – Delete Query (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR36-01	The system shall enable the user to delete a query.	The end-user is successfully able to delete self-initiated existing queries in the system

FR37: User – Edit User Profile (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR37-01	The system shall enable the user to insert new details in the user fields.	The end-user is successfully able to update his own profile in the system
FR37-02	The system shall enable the user to save changes made in the user data.	

FR38: User – Send Messages (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR38-01	The system shall enable the user to send textual messages in chat with the counsellor.	The end user is successfully able to send messages to required entities in the system
FR38-02	The system shall enable the user to send documents such as images and pdfs in the chat.	

FR39: User – Receive Messages (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR39-01	The system shall enable the user to receive messages in chat with the counsellor.	The end user is successfully able to send/receive messages to required entities in the system
FR39-02	The system shall enable the user to receive documents such as images and pdfs in the chat.	

FR40: User – Voice Chat (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR40-01	The system shall enable the user to communicate with the counsellor using voice chat.	The end user is successfully able to make voice calls to required entities in the system
FR40-02	The system shall enable the user to mute/unmute his/her mic during the call as well.	

FR41: User – Video Chat (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
----------	-------------------------	---------------------

FR41-01	The system shall enable the user to communicate with the counsellor using video chat.	The end user is successfully able to make video calls to required entities in the system
FR42-02	The system shall enable the user to disable/enable his/her camera during the call as well.	

FR42: User – Log out (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR42-01	The system shall enable the user to log out of the system.	The end user is successfully able to log out the system

FR43: Counsellor – Log in (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR43-01	The system shall enable the counsellor to access the sign in page.	The counsellor is successfully able to log in the system
FR43-02	The system shall enable the counsellor to enter his/her username and password.	
FR43-03	The system shall authenticate the provided counsellor credentials.	
FR43-04	The system shall sign in the counsellor in case of correct authentication.	

FR44: Counsellor – View Assigned Requests (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR44-01	The system shall enable the counsellor to view all the assigned requests	The counsellor is successfully able to view all assigned user requests the system
FR44-02	The system shall enable the counsellor to read details of individual requests as well	

FR45: Counsellor – Filter Assigned Requests (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR45-01	The system shall enable the counsellor to apply filters on different parameters given.	The counsellor is successfully able to search/filter all assigned user requests the system
FR45-02	The system shall display the filtered results to the counsellor.	

FR46: Counsellor – Send Messages (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR46-01	The system shall enable the counsellor to send textual messages in chat with the user.	The counsellor is successfully able to send/receive messages to required entities in the system
FR46-02	The system shall enable the counsellor to send documents such as images and pdfs in the chat.	

FR47: Counsellor – Receive Messages (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR47-01	The system shall enable the counsellor to receive messages in chat with the user.	The counsellor is successfully able to send/receive messages to required entities in the system
FR47-02	The system shall enable the counsellor to receive documents such as images and pdfs in the chat.	

FR48: Counsellor – Voice Chat (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR48-01	The system shall enable the counsellor to communicate with the user using voice chat.	The counsellor is successfully able to make voice calls to required entities in the system
FR48-02	The system shall enable the counsellor to mute/unmute his/her mic during the call as well.	

FR49: Counsellor – Video Chat (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR49-01	The system shall enable the counsellor to communicate with the user using video chat.	The counsellor is successfully able to make video calls to required entities in the system
FR49-02	The system shall enable the counsellor to disable/enable his/her camera during the call as well.	

FR50: User – Log out (MUST)

Req. No.	Functional Requirements	Acceptance Criteria
FR50-01	The system shall enable the counsellor to log out of the system.	The counsellor is successfully able to log out the system

3.2 Non-functional requirements

Non-functional requirements are defined as overall characteristics of the system. These are actual constraints and quantitate criteria for the system. These may or may not be from the user directly but are also some underlying quality factors which need to be keep in focus while the development process of a system, these requirements do not have a formal acceptance criterion. The identified non-functional requirements are as follows:

NFR01: Performance

NFR01-01	Average load time of the starting page of the system must be less than 2 second.
NFR01-02	Average processing time taken by the system to complete a transaction/request should be less than 10 seconds.
NFR01-03	System Mean Time to Failure should not be more than 60seconds within 24 hours of use.
NFR01-04	Average system response time should not be greater than 5 seconds.
NFR01-05	System must successfully run on a client machine with 256 MB RAM or above.
NFR01-06	100 Students should be able to simultaneously access the system and update the database.

NFR02: Security

NFR02-01	System must provide access to authorized users only that enter through the login module.
----------	--

NFR02-02	System must not provide access to ANY user EXCEPT the designated user to update the database.
NFR02-03	No user can view data of any other user through any report or views provided by the system.
NFR02-04	After the end of a user Session, no information must be saved anywhere on the client machine.

NFR03: Defects-Maintenance

NFR03-01	Post Release defects of the system must not exceed 1 critical bug per month.
NFR03-02	Post Release bug fixing should not take more than 5 hours.

NFR04: Documentation

NFR04-01	Help documentation must be complete in providing information about each and every module and functionality provided by the system.
NFR04-02	Help option must be easily accessible on all system web pages.
NFR04-03	Help must be written using minimal technical terms; any technical terms used must be additionally defined at the end of the document

NFR05: Disaster Recovery

NFR05-01	In case of client /server crash all information/data should be recoverable within 30 minutes of the incidence.
----------	--

3.3 Use Case Diagram

The use case diagram of the entire system has been made by keeping in view the main functionality of the entire system. The use case diagram signifies all the major roles and functionalities of the system with respect to all the actors involved in the system. The diagram was constructed using “Draw.IO” tool online. The details regarding the use case diagram are as follows:

Actors:

1. Super Admin
2. Admin

3. Refugee / Migrant (End User)
4. Counsellor

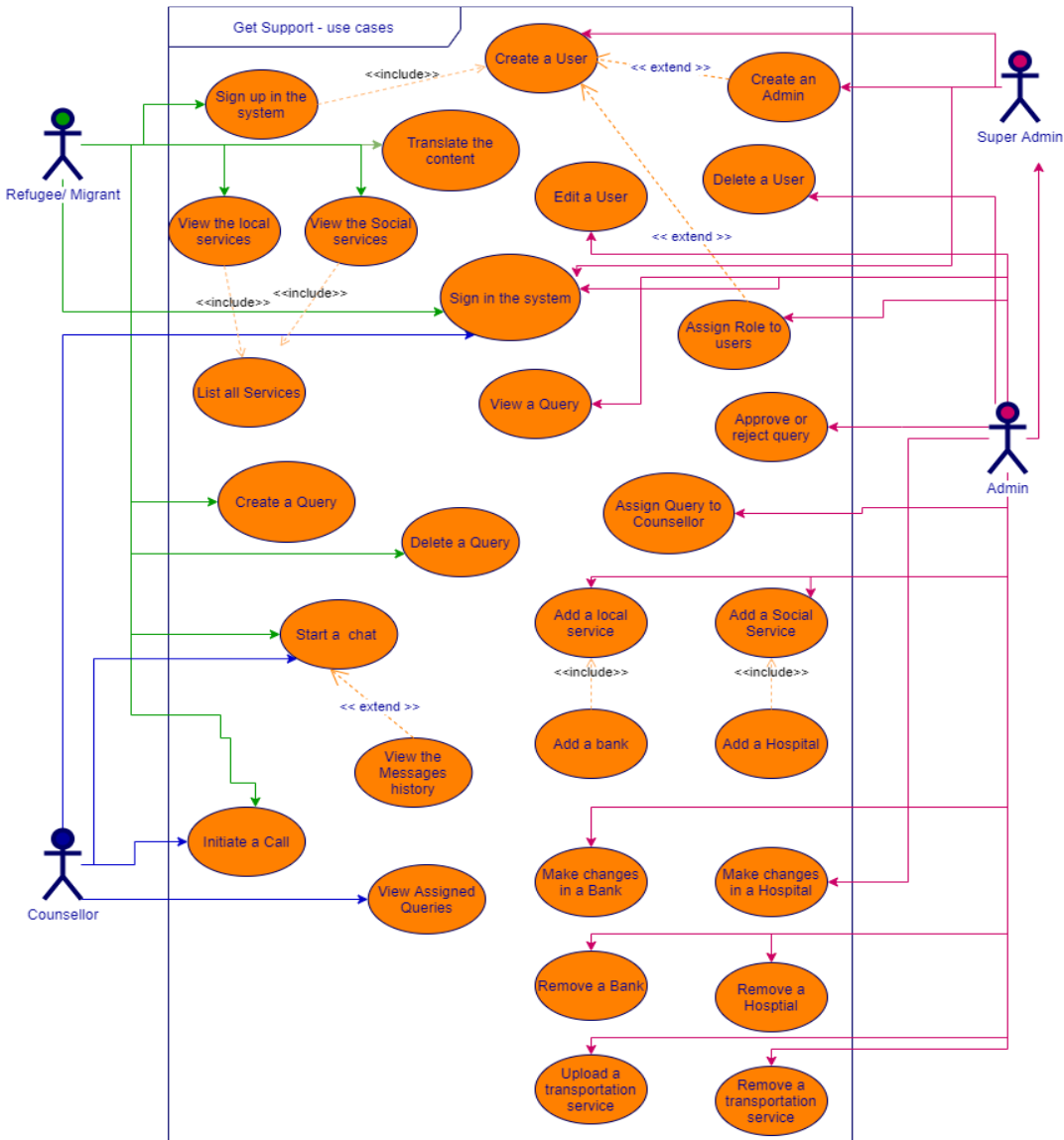


Figure 15 Use Case Diagram

These actors each have specific purposes. A distinct colour is assigned to each type of actor, such that the arrows can be identified from one another and there is no confusion for the readers to actually under and navigate through the diagram. The diagram shows an “extend” link between the “Admin” & “Super Admin”. This has been done to signify that the admin type actor is a child of the main Actor “Super Admin”. This practice is used to group similar types of actors with each other.

3.4 Use Cases Benefits chart

The table below is made an in attempt to summarize all the use cases in the system in a such way that they signify their importance for each type of user in the system (Alexsoft, 2018). This table contains data from the use case descriptions of the system, which will be discussed afterwards.

Services				Benefits			Feasibility	Potential Frequency of use	Comments	Priority
ID	Title	Use Case ID	Description	Refugee / Migrant	Counsellor	Get Support Team				
A	Admin Settings	UC-2	Creating new users	A help seeker may be directly registered into the system by the Admin.	A help provider or professional would be registered in the system	The new user is saved, and a set password email is sent on the email address	Good	High		High
		UC-3	Creating new Roles			The newly registered users can be assigned specific roles, clearly stating their privileges in the system	Good	One off-Only on system deployment	The roles are fixed to only 4	High
		UC-10	Updating existing users	Details are updated if some wrong information was inserted	Details are updated if some wrong information was inserted	The user records data integrity and authenticity are maintained if the records are updated properly	Good	Rarely - user dependent		High
		UC-11	Updating existing roles			The roles permissions may be added or removed to help administer the system better	Good	Rarely - user dependent		High
		UC-20	Deleting existing users			An extra user will be removed from the system	Good	Rarely - user dependent		High

		UC-21	Deleting existing roles			An extra or un-usable role will be removed from the system	Good	Rarely - user dependent		High
		UC-12	Assigning a Counsellor to user query	The Query generated by the User will be assigned a counsellor to deal with.	The counsellor will get the opportunity to help a user	The admin decided rather a query should be approved to assign the counsellor or not. The query may also be rejected	Good	High		High
		UC-6	Creating new Social Services			New social services are added in the system database	Good	Low		High
		UC-7	Creating new departments in existing Social Services.	The departments added may benefit the help seeker			Good	High		High
		UC-4	Creating new Local Services			New local services are added in the system database	Good	Low		High
		UC-5	Creating new departments in existing Local Services	The departments added may benefit the help seeker			Good	High		High

		UC-8	Updating Local Services and its departments	The details of the departments are updated			Good	High		High
		UC-9	Updating Social Services and its departments	The details of the departments are updated			Good	High		High
B	Refugee/ Migrant Settings	UC-13	Translating the content	The page content is translated into the selected language			Good	High		High
		UC-14	Creating a Query	The query is created for any problem faced by the user and assigned to admin for review		The admin decided rather a query should be approved to assign the counsellor or not. The query may also be rejected	Good	High		High
		UC-15	Updating a Query	The query is updated, according to the user requirements			Good	Low		High
		UC-16	Deleting a Query	The query may be deleted if the user desires it to be deleted			Good	Low		High

		UC-17	Initiating the chat	The user may be able to initiate the chat with the assigned counsellor	The counsellor will get the opportunity to help a user		Good	High		High
		UC-18	Initiating an Audio/Video Call	The user may be able to initiate the call with the assigned counsellor	The counsellor will get the opportunity to help a user		Good	Medium		High
C	Counsellor Settings	UC-19	Responding to assigned queries		The counsellor may be view the assigned queries and respond to them via chat using UC-18 & UC-19		Good	High		High

Table 5 Use Case Benefits Analysis

3.5 Use case Descriptions

A use case description states the written and formal methods and stepwise executions of user actions on the system. This system contains many kinds of use-cases and the modelling of these use-cases required the utmost precision. This written description of use cases also enhances the scope of the functional requirements identified during the requirements phase. Each use case description contains different parameters and according to a certain use case description, these parameters are changed. The identified use case descriptions of the system are as follows:

Use Case ID:	UC-1		
Use Case Name:	Singing-in of Users	Version No:	1
End Objective:	The user is logged in to the system		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	Super Admin, Admin, Refugee/immigrant, Counsellor		
Trigger:	All defined Actors		
Frequency of Use:	Very High		
Preconditions			
1. The system presents the login page with all the required fields.			
Basic Flow			
Step	User Actions	System Actions	
1	The user inserts his/ her username	The system shows the written username on the username field	
2	The user inserts his/ her password.	The system hides the written password.	
3	The user clicks submit button.	The system authenticates the credentials and logs the user in.	
Alternate Flow			
Step	User Actions	System Actions	

1	The user has 'keep signed in' checked and accesses the website directly.	The system authenticates the credentials and logs the user in directly without asking for the credentials
Exception Flow		
1	The user inserts his/ her username	The system shows the written username on the username field
2	The user inserts his/ her wrong password.	The system hides the written password.
3	The user clicks submit button	The system authenticates the credentials and brings the user back to log in page with error message.
Post conditions		
1. The user is logged in to his/her dashboard.		
NFRS		
Security		

Use Case ID:	UC-2		
Use Case Name:	Creating new users.	Version No:	1
End Objective:	A new user is created.		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	Super Admin		
Frequency of Use:	Medium		
Preconditions			
1. The user is logged in and is at the admin panel.			

Basic Flow		
Step	User Actions	System Actions
1	The user inserts the new user. Email/username	The system shows the written username on the username field
2	The user inserts the new user. First Name and Last Name	The system shows the written first and last name on the name fields
3	The user assigns a specific role to the new user	
3	The user clicks submit button.	The system saves the new user.
Alternate Flow NONE		
Exception Flow		
1	The user inserts a pre-existing email address for the new user	The system shows the written username on the username field
3	The user clicks submit button	The system gives an error saying “username already exists”
Post conditions		
2. The new user is saved, and a set password email is sent on the email address.		

Use Case ID:	UC-3		
Use Case Name:	Creating new Roles	Version No:	1
End Objective:	A new role is created in the system.		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	Super Admin		
Frequency of Use:	Low		
Preconditions			
1. The user is logged in and is at the admin panel.			

Basic Flow		
Step	User Actions	System Actions
1	The user inserts the new role name.	The system shows the written name on the name field.
2	The user inserts the allowed permissions.	
3	The user removes the un allowed permissions.	
3	The user clicks submit button.	The system saves the new role
Alternate Flow NONE		
Exception Flow NONE		
Post conditions		
3. The new role is saved and is available to be assigned to new users.		

Use Case ID:	UC-4		
Use Case Name:	Creating new Local Services.	Version No:	1
End Objective:	A new local service is saved		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	Super Admin, Admin		
Frequency of Use:	Medium		
Preconditions			
1. The user is logged in and is at the admin panel.			

Basic Flow		
Step	User Actions	System Actions
1	The user inserts the new local service name in the name field.	The system shows the written name in name field.
2	The user inserts all other fields in the input fields.	
3	The user clicks submit button.	The system saves the new local service.
Alternate Flow NONE		
Exception Flow NONE		
Post conditions		
1. The new local service is saved and is available for usage in the entire system.		

Use Case ID:	UC-5		
Use Case Name:	Creating new departments in existing Local Services.	Version No:	1
End Objective:	A new department is created under an existing local service.		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	Super Admin, Admin		
Frequency of Use:	Medium		
Preconditions			
1. The user is logged in and is at the admin panel. 2. A local service is created.			
Basic Flow			

Step	User Actions	System Actions
1	The user inserts the new department name.	The system shows the written name in name field.
2	The user selects the existing local services from the selection menu.	
3	The user clicks submit button.	The system saves the new department in local services.
Alternate Flow		
1	The user selects the local service from the selection	
2	The user inserts the new department name.	The system shows the written name in name field.
3	The user clicks submit button.	The system saves the new department in local services.
Exception Flow NONE		
Post conditions		
<ol style="list-style-type: none"> 1. The new department is associated local service. 2. The new department is saved in the system. 3. The new department is available for display to all the end users. 		

Use Case ID:	UC-6		
Use Case Name:	Creating new Social Services.	Version No:	1
End Objective:	A new social service is saved		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	Super Admin, Admin		
Frequency of Use:	Medium		
Preconditions			
2. The user is logged in and is at the admin panel.			
Basic Flow			
Step	User Actions	System Actions	
1	The user inserts the new social service name in the name field.	The system shows the written name in name field.	
2	The user inserts all other fields in the input fields.		
3	The user clicks submit button.	The system saves the new social service.	
Alternate Flow NONE			
Exception Flow NONE			
Post conditions			
2. The new social service is saved and is available for usage in the entire system.			

Use Case ID:	UC-7		
Use Case Name:	Creating new departments in existing Social Services.	Version No:	1
End Objective:	A new department is created under an existing social service.		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	Super Admin, Admin		
Frequency of Use:	Medium		
Preconditions			
1. The user is logged in and is at the admin panel. 2. A social service is created.			
Basic Flow			
Step	User Actions	System Actions	
1	The user inserts the new department name.	The system shows the written name in name field.	
2	The user selects the existing social services from the selection menu.		
3	The user clicks submit button.	The system saves the new department in social services.	
Alternate Flow			
1	The user selects the social service from the selection		
2	The user inserts the new department name.	The system shows the written name in name field.	

3	The user clicks submit button.	The system saves the new department in social services.
Exception Flow NONE		
Post conditions		
4. The new department is associated social service. 5. The new department is saved in the system 6. The new department is available for display to all the end users.		

Use Case ID:	UC-8		
Use Case Name:	Updating Local Services and its departments.	Version No:	1
End Objective:	A local service and its department is updated		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	Super Admin, Admin		
Frequency of Use:	Medium		
Preconditions			
1. The user is logged in and is at the admin panel. 2. The local service is created. 3. A department is created.			
Basic Flow			
Step	User Actions	System Actions	
1	The user updates the new department name.	The system shows the written name in name field.	
2	The user selects a new local service.		
3	The user clicks submit button.	The system updates the records.	

Alternate Flow		
1	The user only updates the core local service data.	The system updates only the core local service data.
Exception Flow NONE		
Post conditions		
<ol style="list-style-type: none"> 1. The record(s) is updated. 2. The updated local service/ department is available for display to all the end users. 		

Use Case ID:	UC-9		
Use Case Name:	Updating Social Services and its departments.	Version No:	1
End Objective:	A social service and its department is updated		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	Super Admin, Admin		
Frequency of Use:	Medium		
Preconditions			
1. The user is logged in and is at the admin panel. 2. The social service is created. 3. A department is created.			
Basic Flow			
Step	User Actions	System Actions	
1	The user updates the new department name.	The system shows the written name in name field.	
2	The user selects a new social service.		

3	The user clicks submit button.	The system updates the records.
Alternate Flow		
1	The user only updates the core social service data.	The system updates only the core social service data.
Exception Flow NONE		
Post conditions		
3. The record(s) is updated. 4. The updated social service/ department is available for display to all the end users.		

Use Case ID:	UC-10		
Use Case Name:	Updating existing users.	Version No:	1
End Objective:	A user is updated		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	Super Admin, admin		
Frequency of Use:	Medium		
Preconditions			
1. The user is logged in and is at the admin panel. 2. A user record is created other than the logged in user.			
Basic Flow			
Step	User Actions	System Actions	
1	The user updates the user. Email/username	The system shows the written username on the username field	
2	The user updates the new user. First Name and Last Name	The system shows the written first and last name on the name fields	
3	The user assigns a specific role to the new user		

4	The user clicks submit button.	The system updates the user.
Alternate Flow NONE		
Exception Flow		
1	The user inserts a pre-existing email address for the new user	The system shows the written username on the username field
2	The user clicks submit button	The system gives an error saying “username already exists”
Post conditions		
1. The user is updated.		

Use Case ID:	UC-11		
Use Case Name:	Updating existing roles.	Version No:	1
End Objective:	A user is updated		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	Super Admin, admin		
Frequency of Use:	Medium		
Preconditions			
1. The user is logged in and is at the admin panel. 2. A role record is created.			
Basic Flow			
Step	User Actions	System Actions	

1	The user updates the name of the role.	The system shows the written username on the username field
2	The user updates the permissions in a role.	
3	The user clicks submit button.	The system updates the role.
Alternate Flow NONE		
Exception Flow NONE		
Post conditions		
1. The role is updated		

Use Case ID:	UC-12		
Use Case Name:	Assigning a Counsellor to user query	Version No:	1
End Objective:	A query is created		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	Super Admin, admin		
Frequency of Use:	High		
Preconditions			
1. The user is logged in and is at the admin panel. 2. A query is created by the user. 3. The query has not been assigned a counsellor.			
Basic Flow			
Step	User Actions	System Actions	

1	The user views the query.	
2	The user assigns the relevant counsellor to the query.	
3	The user clicks submit button.	The system updates the records, and notifies the counsellor through email notification
Alternate Flow		
2	The user rejects the query and does not assign a counsellor	The system notifies the user of the query meeting request rejection.
Exception Flow NONE		
Post conditions		
1. The counsellor and user are notified of it.		

Use Case ID:	UC-13		
Use Case Name:	Translating the webpage	Version No:	1
End Objective:	The content of the website is translated		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	Super Admin, admin, refugee/ asylum seeker, counsellor		
Frequency of Use:	High		
Preconditions			
1. The user is on the website.			
Basic Flow			
Step	User Actions	System Actions	

1	The user selects a language from the language selection.	The system translates the entire content of the website to the selected language.
Alternate Flow NONE		
Exception Flow NONE		
Post conditions		
1. The content of the website is translated.		

Use Case ID:	UC-14		
Use Case Name:	Creating a Query	Version No:	1
End Objective:	A query is created		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	refugee/ asylum seeker		
Frequency of Use:	High		
Preconditions			
1. The user is on the website.			
Basic Flow			
Step	User Actions	System Actions	
1	The user accesses the query creation page.		
2	The user fills out all required fields.		
3	The user submits the query.	The system saves the query and notifies admin of a new query.	
Alternate Flow NONE			
Exception Flow NONE			

Post conditions
1. The query is created and saved.

Use Case ID:	UC-15		
Use Case Name:	Updating a Query	Version No:	1
End Objective:	A query is updated		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	refugee/ asylum seeker		
Frequency of Use:	Low		

Preconditions
1. The user is on the website. 2. A query is created.

Basic Flow		
Step	User Actions	System Actions
1	The user accesses the query page.	
2	The user updates the required fields of the query	
3	The user submits the query.	The system updates the query and notifies admin.

Alternate Flow NONE
Exception Flow NONE
Post conditions
1. The query is updated.

Use Case ID:	UC-16		
Use Case Name:	Deleting a Query	Version No:	1
End Objective:	A query is updated		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	refugee/ asylum seeker		
Frequency of Use:	Low		
Preconditions			
3. The user is on the website. 4. A query is created.			
Basic Flow			
Step	User Actions	System Actions	
1	The user accesses the query page.		
2	The user clicks to delete the query.	The system asks to confirm again before deleting.	
3	The user confirms deletion	The system deletes the query.	
Alternate Flow NONE			
Exception Flow NONE			
Post conditions			
1. The query is deleted.			

Use Case ID:	UC-17		
Use Case Name:	Initiating a chat	Version No:	1
End Objective:	A chat is initiated between users		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	refugee/ asylum seeker, counsellor		
Frequency of Use:	High		
Preconditions			
<div>1. The users are logged in.</div> <div>2. A query is registered.</div> <div>3. Both parties are online.</div>			
Basic Flow			
Step	User Actions	System Actions	
1	The user clicks to start the chat.	The system responds by displaying the relevant party to each other	
2	The user types and sends a message	The system sends the encrypted message to the other side	
Alternate Flow NONE			
Exception Flow NONE			
Post conditions			
<div>1. The chat is initiated</div>			

Use Case ID:	UC-18		
Use Case Name:	Initiating an Audio/Video Call	Version No:	1
End Objective:	A call is initiated between users		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	refugee/ asylum seeker, counsellor		
Frequency of Use:	High		
Preconditions			
4. The users are logged in.			
5. A query is registered.			
6. Both parties are online.			
Basic Flow			
Step	User Actions	System Actions	
1	The user clicks to start the chat.	The system responds by displaying the relevant party to each other	
2	The user clicks on the call icon to start the call	The system sends the other party notification of incoming call.	
3	The call is started	The system connects the VOIP or video RTC session.	
Alternate Flow			
2	The other user rejects the call.	The system notifies the caller of call rejection and call is not established.	
Exception Flow NONE			
Post conditions			
2. The call is initiated			

Use Case ID:	UC-19		
Use Case Name:	Responding to assigned queries	Version No:	1
End Objective:			
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	counsellor		
Frequency of Use:	High		
Preconditions			
<div>1. The users are logged in.</div> <div>2. A query is registered and assigned to counsellor.</div>			
Basic Flow			
Step	User Actions	System Actions	
1	The user clicks on the view all assigned quires.	The system responds by displaying the relevant information.	
Alternate Flow NONE			
Exception Flow NONE			
Post conditions			
<div>3. The queries are listed</div>			

Use Case ID:	UC-20		
Use Case Name:	Deleting existing users.	Version No:	1
End Objective:	A user is deleted		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21

Actor(s):	Super Admin, admin	
Frequency of Use:	Medium	
Preconditions		
3. The user is logged in and is at the admin panel.		
4. A user record is created other than the logged in user.		
Basic Flow		
Step	User Actions	System Actions
1	The user selects the user to delete	The system shows the confirm dialog box
2	The user confirms the delete option	The system removes the user record from the database.
Exceptional Flow NONE		
Alternate Flow		
2	The user denies deleting the record	The system does not remove the user from the database/
Post conditions		
2. The user is deleted.		

Use Case ID:	UC-21		
Use Case Name:	Deleting existing roles.	Version No:	1
End Objective:	A user is deleted		
Created by:	Abubaker Omar	On (date):	4/16/21
Last Update by:	Abubaker Omar	On (date):	4/17/21
Approved by:	Abubaker Omar	On (date):	4/17/21
Actor(s):	Super Admin, admin		
Frequency of Use:	Medium		
Preconditions			
5. The user is logged in and is at the admin panel.			

6. A role record is created other than the logged in user.		
Basic Flow		
Step	User Actions	System Actions
1	The user selects the role to delete	The system shows the confirm dialog box
2	The user confirms the delete option	The system removes the role record from the database.
Exceptional Flow NONE		
Alternate Flow		
2	The user denies deleting the record	The system does not remove the role from the database/
Post conditions		
3. The role is deleted.		

4 Implementation

4.1 System Diagrams

The system has a different system diagram associated with it. For this project, the most relevant system diagrams have been included and explained – The entity Relationship diagram and Sequence diagram.

4.2 Entity Relationship Diagram (ERD)

This diagram is used to describe the database schema graphically. The ERD contains all the databases tables created and all the fields mapping as well. The system's ERD is in two different versions: one is the ideal version which was constructed keeping the future goals and scalability in mind, and the other one is the actual ERD generated with the actual database tables used in the current version of the system. Due to time constraints and skill hampering in this region, the used database schema is not in the most efficient form, however, it is being used in the system without any issues.

4.3 Sequence Diagram

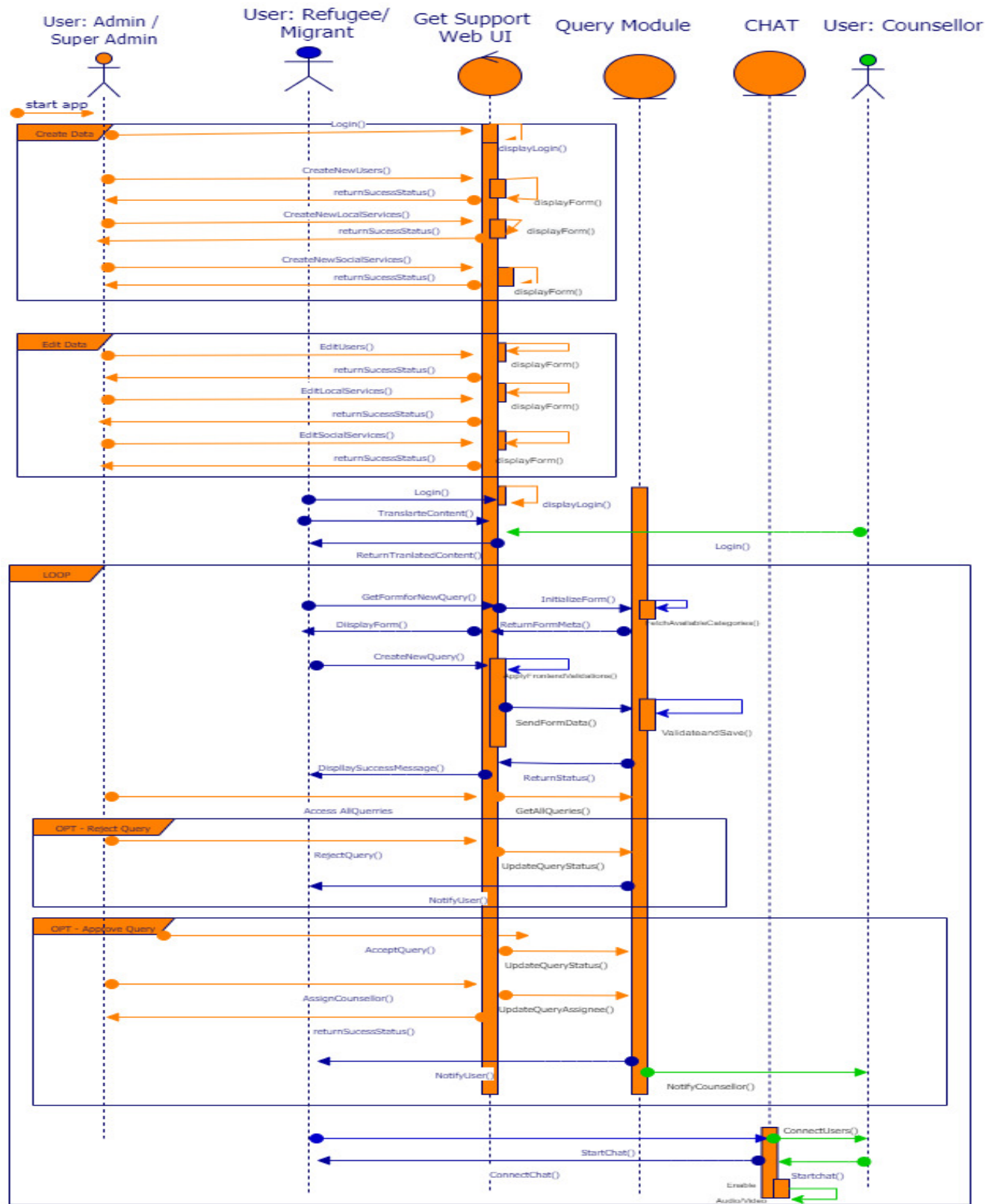


Figure 17 Sequence Diagram

4.4 Sequence Diagram Explanation

A sequence diagram captures the interaction between different entities in a system. It is a behavioural diagram, and it focuses on the behaviour of the system in a sequential flow of events (Visual Paradim, 2020). These events can be triggered by a user or by message exchange between different modules of the system. The reason for using a sequence diagram for this project is based on the streamlined flow of events. The system follows a very straightforward path for execution, and this is modelled very efficiently on the sequence diagram. The normal flow of events in the system is:

1. Admin or Super Admin creates new services (Local & Social)
2. Admin or Super Admin creates new users and roles.
3. An end-user (Refugee/ Migrant) accesses the website.
4. The end-user uses the translation module to understand the content.
5. The end-user logs in and creates a new query.
6. Admin views the query and either accepts it or rejects it.
7. If the Admin approves the user query, he/she assigns the query to one of the available counsellors.
8. The end-user and counsellor are notified of the assignment.
9. The Counsellor and end-users can now communicate with each other using the chatting module capable of providing Audio and Video calling features as well.
10. If the Admin rejects the query, then the user is notified, and no further action is performed on the query.

All these steps are modelled on the sequence diagram effectively, such that each user activity and the system response can be clearly understood from it. The sequence diagram provided insight into the actual flow of events in the system. The initial versions of the sequence diagram were not stable, and some functionality conflicted with the existing modules. The attached diagram focuses on the core modules of the system and the core actions performed on the system, highlighting only the important processes of the system.

5 System Code Explanation

The system was developed using the “Python Django framework”, the rationale for using this specific framework is discussed in detail in the previous sections of this report. The first thing to discuss is the architecture of the project – it follows the main architecture of Django, which is the MTV (Model Template View) architecture. This architecture is based on the traditional MVC (Model View Controller) architecture style. Comparing it to the MVC format, MTV is more intuitive than the traditional approach. The details of each part of the architecture are described below.

5.1 MTV

The following section will detail the elements of the Model View Controller (MTV) architecture style.

5.1.1 Model:

The Model is the database structure or schema; each database table is considered a *class* that extends the Model class built-in Django. The Models make it very easy to develop a database schema and turn the logical schema into the physical schema on the DBMS (Database Management System). The database server needs to be linked with the Django project and this handles the database settings and other

functions on its own. There are many different Models in our project, and they can be inspected with the ERD diagram we have discussed before. Let us take one sample Model from my project for discussion purposes.

```
CATEGORY_CHOICES = (
    ("Emergency_Support", "Emergency_Support"),
    ("Fix_Meeting", "Fix_Meeting"),
    ("Others", "Others"),
    ("Direct_Meeting", "Direct_Meeting"),
)
STATUS_CHOICES = (
    ("PENDING", "PENDING"),
    ("APPROVED", "APPROVED"),)

class Query(models.Model):
    sno = models.AutoField(primary_key=True,blank=True)
    currenstatus = models.BooleanField(default=True)
    category_type =
models.CharField(max_length=50,choices=CATEGEORY_CHOICES,default=True,
blank=True)
    details = models.TextField(default=True,blank=True)
    user = models.ForeignKey(User,on_delete=models.CASCADE,null=True,
blank=True)
    time = models.DateTimeField(auto_now_add=True)
    status = models.CharField(default="PENDING",max_length=50)
    assign = models.CharField(max_length=50)

    def __str__(self):
        return self.details
```

The sample Model is of the Query Table. This Model has eight distinct attributes. All the attributes have a proper data type assign to them. Django will treat the 'Query' class as a table for the database and it will treat all the variables in it as the fields or columns of the database concerning the datatype assigned to it.

5.1.2 Template

The template contains the frontend part of the system; these contain the HTML code, and the HTML is linked with the CSS and Java-script files. The templates are unique as in Django they contain the HTML, but they also have a template language as well. There are many different template tags available in Django and they all are focused on rendering the database contents on the front end in a dynamic order. There also exists a phenomenon known as template inheritance, which simply states to put the reusable code of the entire system in one template and other templates can simply extend it and put in more content. This helps in the code reusability as well as provide more freedom to make changes, as necessary. A simple template is pasted below.

```

{% extends 'admin_base.html' %}
{% load widget_tweaks %}
{% block content %}

<head>
  <style media="screen">
    a:link {
      text-decoration: none;
    }

    .note {
      text-align: center;
      height: 80px;
      background: -webkit-linear-gradient(left, #0072ff, #8811c5);
      color: #fff;
      font-weight: bold;
      line-height: 80px;
    }

    .form-content {
      padding: 5%;
      border: 1px solid #ced4da;
      margin-bottom: 2%;
    }

    .form-control {
      border-radius: 1.5rem;
    }

    .btnSubmit {
      border: none;
      border-radius: 1.5rem;
      padding: 1%;
      width: 20%;
      cursor: pointer;
      background: #0062cc;
      color: #fff;
    }

    .menu {
      top: 50px;
    }
  </style>

  <link
href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
  <script

```

```

src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
  <script
src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
</head>
<br><br><br><br><br><br>
<form method="post" enctype="multipart/form-data">
  {% csrf_token %}
  <div class="container register-form">
    <div class="form">
      <div class="note">
        <p>ADD Users</p>
      </div>
      <div class="form-content">
        {% render_field userForm.first_name class="form-control"
placeholder="First Name" %}<br>
        {% render_field userForm.last_name class="form-control"
placeholder="Last Name" %}<br>
        {% render_field userForm.username class="form-control"
placeholder="Username" %}<br>
        {% render_field userForm.password class="form-control"
placeholder="Password" %}<br>
        {% render_field customerForm.email class="form-control" placeholder="Email" %}<br>
        {% render_field customerForm.mobile class="form-control" placeholder="Mobile" %}<br>
        <button type="submit" class="btnSubmit">ADD</button>
      </div>
    </div>
  </div>
</form>

{% end block content %}

```

This is the template for the Add Users form, it extends a base template that holds the Navigation bar and other useful imports which remain the same throughout the project. The rest of the template contains the HTML components and the required template tags for the functionality to be successful.

5.1.3 View

The view corresponds to the controller of the traditional MVC architecture. The view contains the actual logic of the data manipulation, it is simply a function that receives the user input and processes the request and then returns a response. It can also be thought of as an API endpoint as well. The requests from the frontend of the website or system hit the endpoint (view), the view does some functionality on the provided input and then returns a response, in Django the response is normally an HTTP response containing a status code, or it can be a JSON response as well. Views in Django are of two types, one is known as class-based views and the other known as function-based views, the class-based views are the

best use for rapid development and are normally used when making REST services. The function-based views are rawer and more intuitive, the developer has the freedom to follow any type of coding practice and structure one wants to. This system also contains function-based views, as they offer more freedom and are more suited for a traditional Django Application. A simple view from the system is as follows:

```
def contact_us(request):
    if request.method == 'POST':
        form = Contact(request.POST, files=request.FILES)
        if form.is_valid():
            form.save()
            messages.success(request, 'Details Submitted')
            return redirect('/')
    else:
        form = Contact()
    return render(request, 'contact.html', {'form': form})
```

This is a view for the Get or Post requests regarding the 'contact us form on the website. The first thing to note is that this view and all views would take the request object as an argument. The first line inside the project checks what type of request is it. If the request is Post then it means the user is going to be inserting some sort of data, the rest of the code inside the Post check focuses on verifying that the data is in the right format or not, if the data is valid the entire contact us form is saved. The else part is considered to be a Get request, in this case, the entire meta-information about the form as well as which HTML page to return to the user is sent as a response.

5.2 Code Distribution

The project code follows a very simplistic structure, taking full advantage of the Django apps classification. The system currently has 2 major apps in it. One is Known as 'chat' and the other as 'app'. The chat app contains the code and details as well as the models for the chat, the other one is the general application for the entire system. This approach can be modified but due to time and skill constraints the current development build was focused only on these two applications, but a better approach could be to make separate apps for separate user functionalities. The overall structure of the project is posted below:

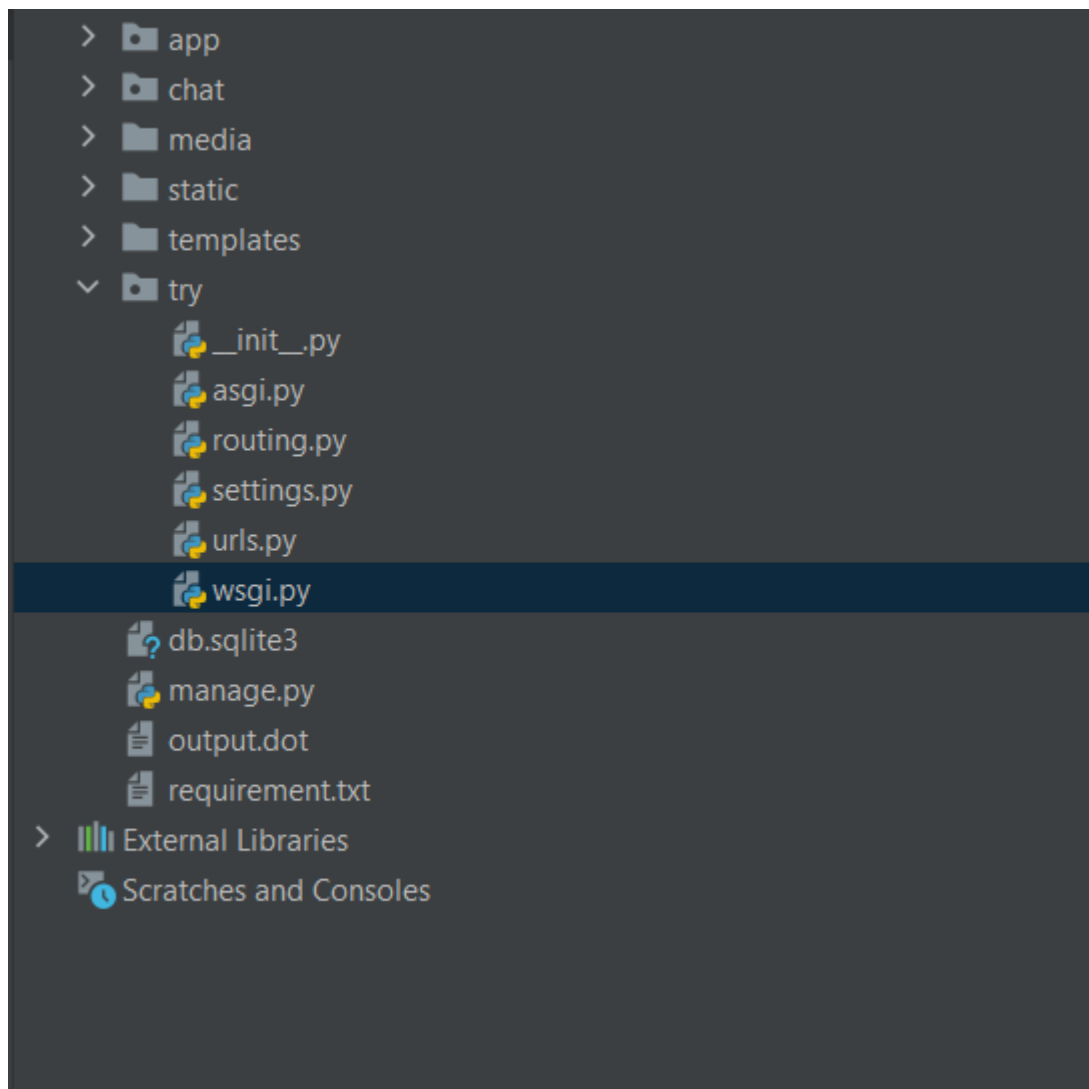


Figure 18 Project Structure

The first two folders are apps and the 'media' folder stores user-uploaded images and files. The 'static' folder holds the static files to be used for the templates, such as images or CSS or Java-Script files. The 'templates' store the front end of the system. The other python files are concerned with the technical project setup and are Django generated. The project's focus is on the translation and the chat module, so the coding details are discussed for these two functionalities only.

5.3 Translation Module

The translation module is one of the most central functionalities of this system. The tool we used to achieve this was by incorporating the 'Google Translation API'. This is a utility developed by google for translation purposes. It holds valid translation capabilities for about 108 languages in many different

dialects. The translation API is given in JavaScript code format as the content is already rendered on the frontend and it's simply translated there. The Js files responsible for the translation are linked inside the base.html template.

```
<Span><input type="text" placeholder="Search.." name="search">
  <button type="submit">Submit</button></Span>
</form>
</div>
</div>
<script src = "{% static 'assets/js/translate/translate.js'%}"></script>
<script src = "{% static 'assets/js/translate/translate2.js'%}"></script>
<div id="translation" style = "float: left; margin : 0; padding : none;" class = "active"></div>
<br>
```

Figure 19 Translation API linkage

```
function googleTranslateElementInit() {
  new google.translate.TranslateElement({pageLanguage: 'en', layout:
google.translate.TranslateElement.InlineLayout.SIMPLE},
'translation');
}
```

This JavaScript function is responsible for getting the entire content of the page and then translating it to the selected language. The functionality is very optimized, the way it works is that it constructs an entire graph of the page content and as soon as there is a change in the selected language the entire graph is traversed and the content changes. The user does not have to make any effort to accomplish this at all. The user simply selects a new language from the dropdown.

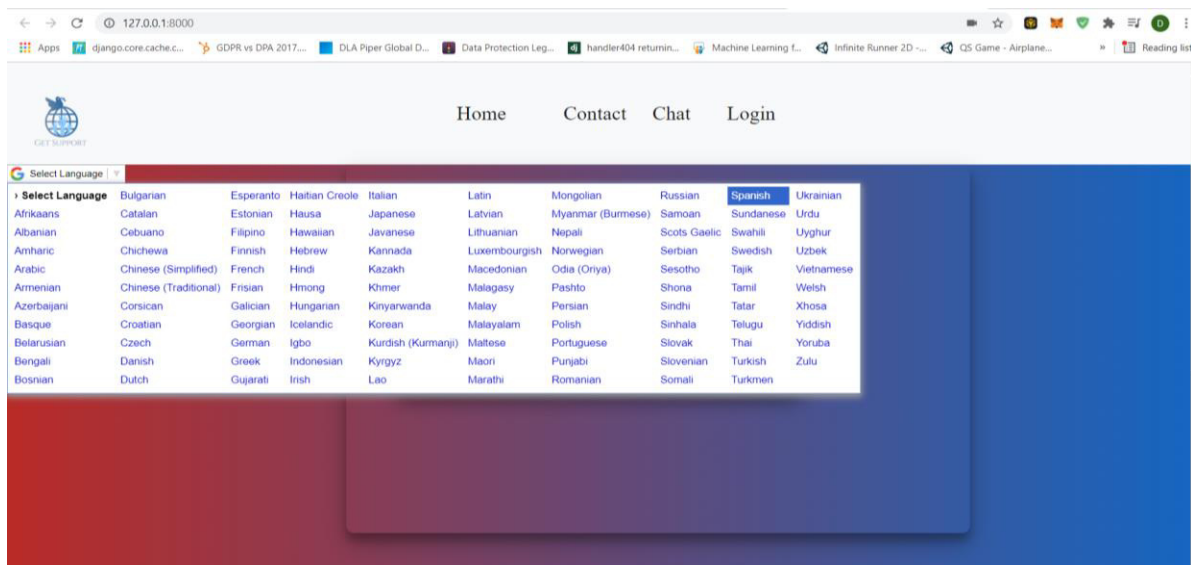


Figure 20 Home Page Translation menu

The language selection gives the user the freedom to easily select any language wants to. The current user selection in the image posted above is 'Spanish', the translated results are pasted below.

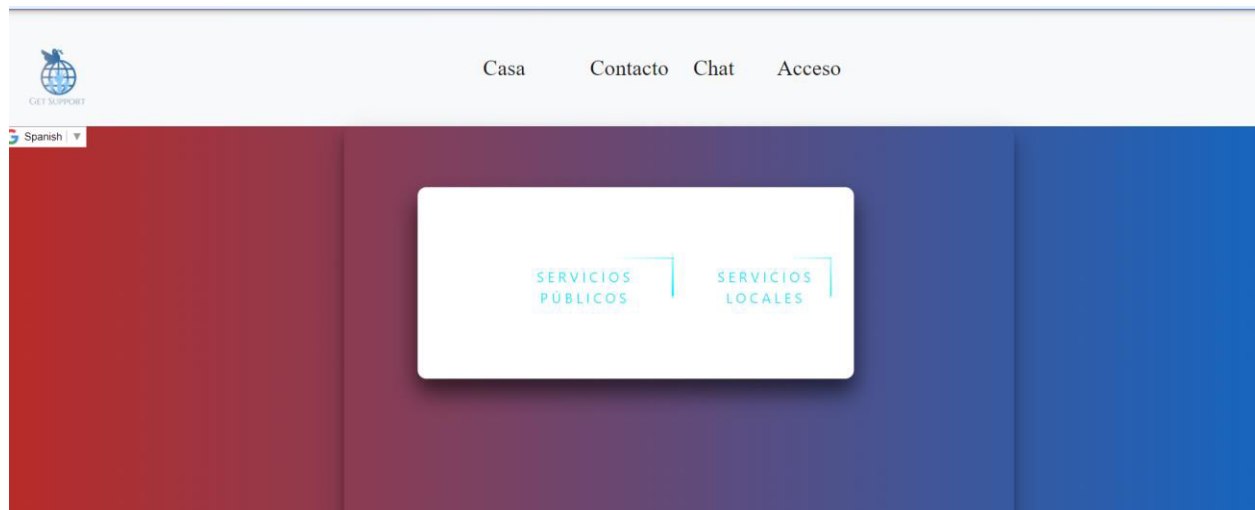


Figure 21 Home Page translated in Spanish.

This translation happens in a matter of micro-seconds, hence to the user, it seems to be instantaneous. The entire website's content is translated, and this change is reflected in other pages as well.



Figure 22 Emergency Chart Translated in Spanish

Just like the emergency chart, all the pages of the website are translated.

5.4 Admin Dashboard

The Admin dashboard is a very important module of the system, this module can be considered as the single source of data in the entire system. This dashboard is responsible for all the website admin-based operations. Our system consists of 2 Kinds of Admins. The first one is the super admin; this type of admin has the master access and permissions to do practically anything with the system itself and make changes directly to the database structure of the system. The super admin right is only reserved for the maintainers or the owner of the system. The system administration is performed through the simple Admin. The difference is that the Admin has Create, Read, Update & Delete rights for the system but this Admin cannot make meta-type changes for the database structure. We have two different types of dashboards for these two different kinds of users. The first one is the Super Admin panel. The Super Admin panel is using the Django Admin interface and it has multiple extra features such as roles and permissions management and history recording.

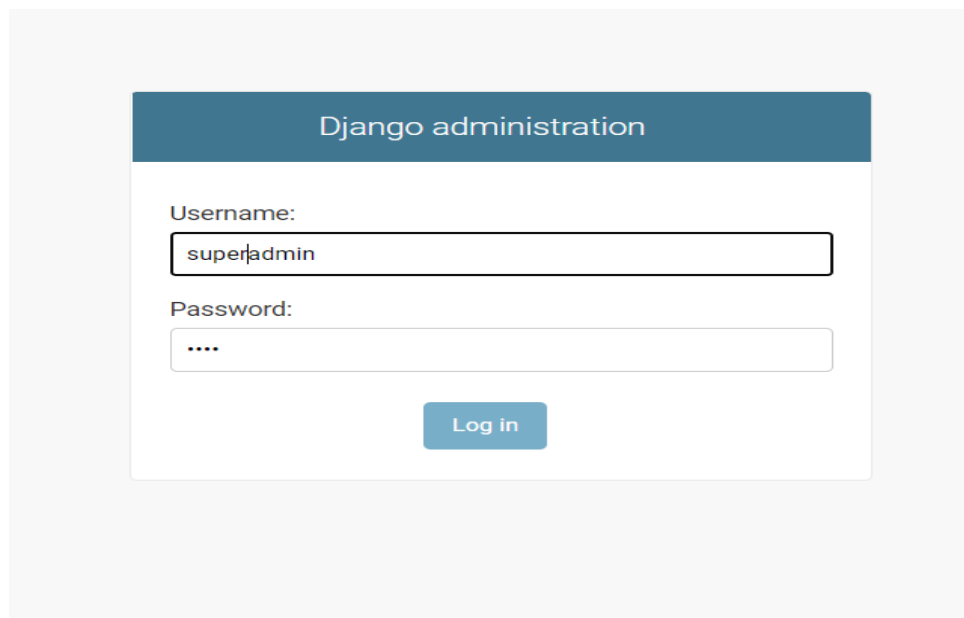


Figure 23 Django Super Admin login Interface

The login panel is also just reserved for the super Admin only, this is also using Django's built-in Admin interface authentication module. After the successful authentication of the Super Admin, the dashboard is accessible.

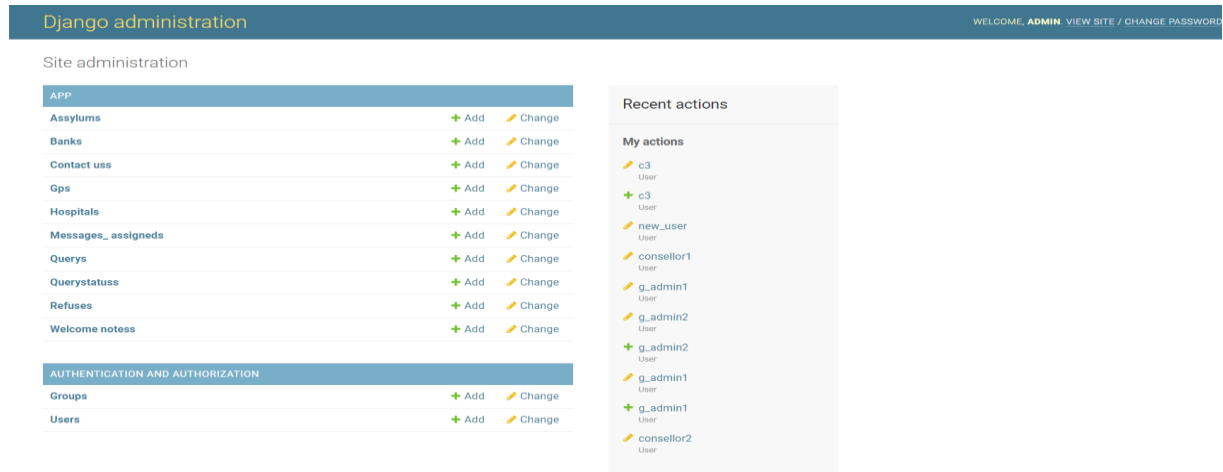


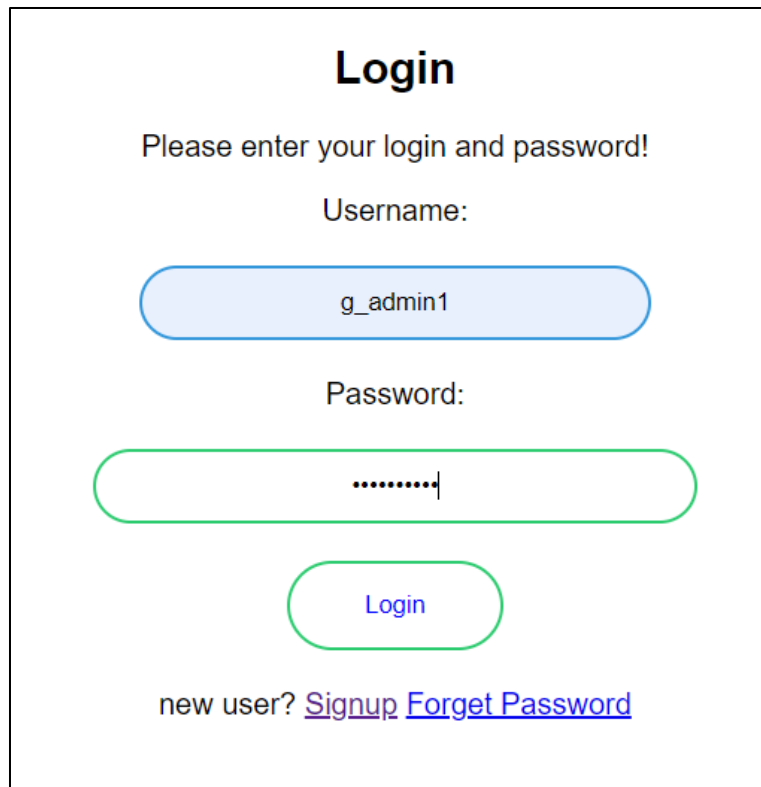
Figure 24 Django Super Admin Interface

This interface is used primarily to set up roles in the system and make normal admins for the efficient administration of the system. The way to make the Super Admin itself on Django is quite straightforward. It is simply a one-line command and is available to be used as soon as the Django project is initialized in the system. In the project folder, the following command is used to create the Super Admin.

```
Python manage.py createsuperuser
```

This command alone would make the Super Admin with all the necessary permissions and will also set up the authentication for this user as well.

The Admin login and dashboard are different from this one. The Admin dashboard contains rights and information about system content.



The login form is titled "Login" in bold black text. Below the title, it says "Please enter your login and password!". There are two input fields: "Username:" with a light blue rounded rectangle containing the text "g_admin1", and "Password:" with a light green rounded rectangle containing seven dots and a cursor. Below these fields is a green rounded rectangle button with the text "Login" in blue. At the bottom, it says "new user? [Signup](#) [Forget Password](#)".

Figure 25 Login Form for Admin and other users

The sign-in is common for all users as well, depending upon the type of user, the dashboard is automatically presented. The Admin dashboard is custom developed, and all the backend and front-end utilities were written by the project developer.

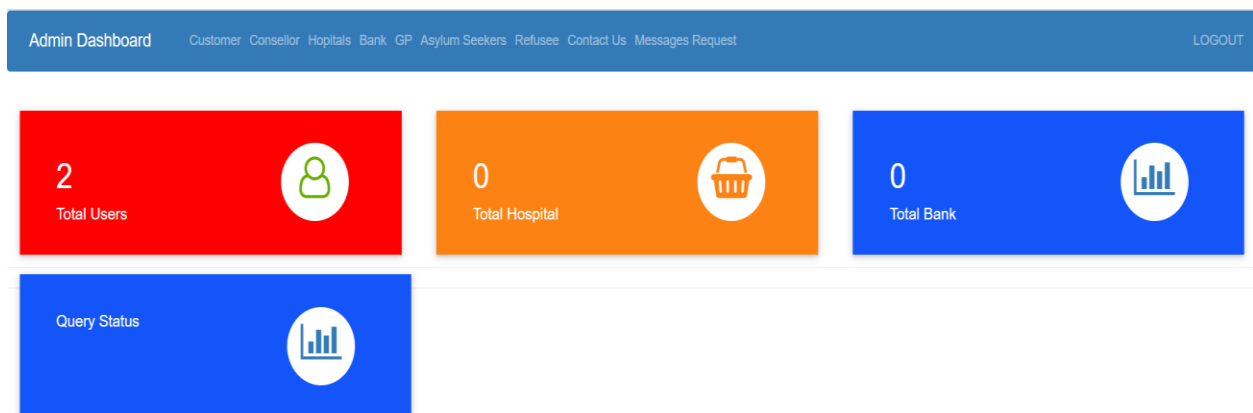


Figure 26 Admin Dashboard

This dashboard features all the administrative controls for the system. There are many different features and access rights in this dashboard, but here only the most important ones are explained. We can take a test study for the insertion of the hospitals.

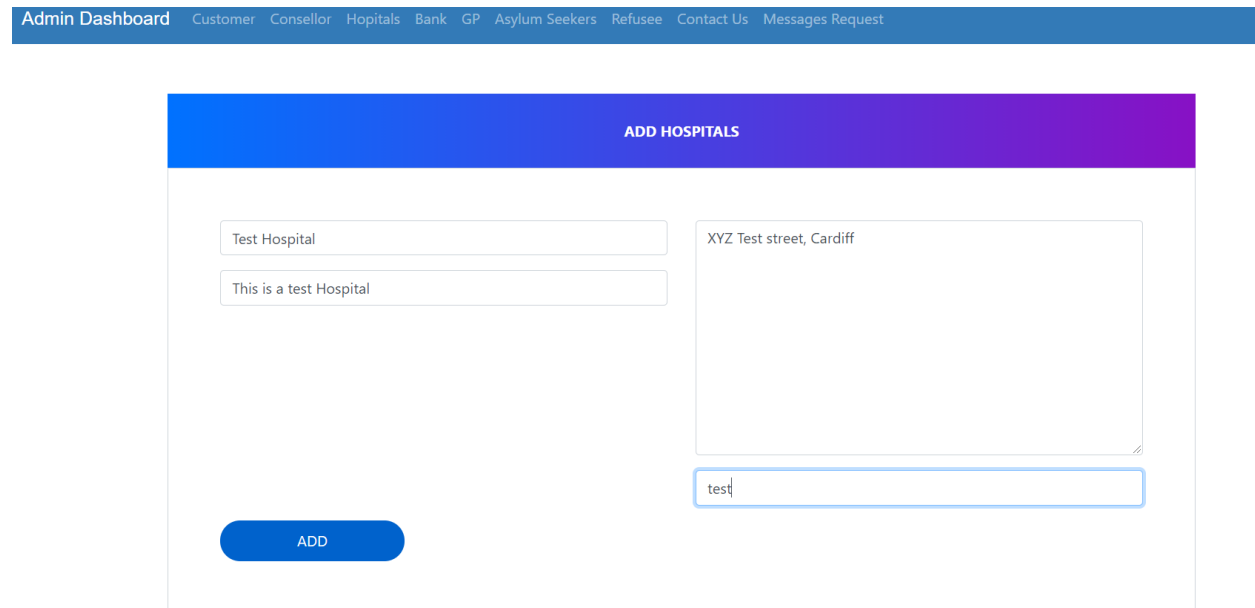


Figure 27 Add hospital Form

This is the hospital insertion form, the admin can add new hospitals in the system from here and it is very easy to do so, once a hospital is added here, it will appear on the website for viewers as well.

```
@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def add_hospital(request):
    hospitalForm=HospitalForm()
    if request.method=='POST':
        hospitalForm=HospitalForm(request.POST, request.FILES)
        if hospitalForm.is_valid():
            hospitalForm.save()
            return HttpResponseRedirect('admin-hospital')
    return
render(request, 'admin_add_hospitals.html', {'hospitalForm':hospitalForm
})
```

This is the view that handles the insertion of the hospital into the system. It has 2 decorators on top of it ensuring that the user making the request is an Admin and is logged in. This is the only case study for View that will be discussed in this report because others are similar to it as well and the detailed explanation of the admin dashboard code would take the focus away from the main content of the report.

5.5 Query Module

This module deals with the creation of query or requests from the end-user (refugee/ migrant). The query creation form is on the website and is very simplistic.

The screenshot shows a web application interface with a navigation bar at the top containing links: Home, Contact, Chat, Profile, Query, All Query, and Logout. Below the navigation bar is a 'Query Form' modal. The form has a blue header with the title 'Query Form'. It contains two main sections: 'Category type:' with a dropdown menu showing 'Fix_Meeting', and 'Details:' with a text area containing the text 'I would like to have a meeting regarding my health insurance'. At the bottom left of the form is a blue 'SEND' button.

Figure 28 Query Creation Form

The corresponding view that handles the form submission is very straightforward.

```
def Queryform(request):
    if request.method == "POST":
        form = QueryForm(request.POST)

        if form.is_valid():
            query = form.save(commit=False)
            query.user = request.user
            query.save()
            return redirect('/')
        form = QueryForm()
    return render(request, "queryform.html", {"form": form})
```

The next half of this process happens at the admin portal. The admin can see the listing of the available user queries. The admin approves or reject the query and simply assign a counsellor the request. This is handled by a complex logic that checks all the auth rights of the admin and the available counsellors and then ensures that a counsellor can only see the requests assigned to him/her.

The same case is for the user, the user should only see the counsellor assigned to him and should be able to chat with only a chat person.

Figure 29 Query Approval Form

This figure shows the query approval form, the admin can also choose to reject the request as well. The name of the user has been kept hidden for privacy purpose in the report.

Emergency_Support	HLEPP	May 12, 2021, 10:12 a.m.	APPROVED	c3	Send	
Fix_Meeting	I need a new meeting	May 13, 2021, 7:38 a.m.	APPROVED	c3	Send	
Emergency_Support	I AM NEW USER	May 13, 2021, 7:46 a.m.	APPROVED	c3	Send	
Direct_Meeting	NEW REQ COUN2	May 13, 2021, 8:18 a.m.	APPROVED	c3	Send	
Emergency_Support	new user 2 query for coun 2	May 13, 2021, 8:24 a.m.	APPROVED	c3	Send	
Others	hello this is other from new user	May 15, 2021, 7:29 a.m.	PENDING	c3	Send	
Emergency_Support	emergnecy for new user	May 15, 2021, 7:57 a.m.	APPROVED	c3	Send	
Fix_Meeting	I would like to have a meeting regarding my health insurance	May 27, 2021, 1:49 p.m.	APPROVED	counselor1	Send	

Figure 30 Request Assignment

The admin has assigned that specific query to a counsellor named counsellor1.



The screenshot shows a web interface for a counsellor's dashboard. At the top, there is a blue circular profile picture placeholder. Below it is a table with 5 entries. The table has columns for an ID, a category, a description, a status, and a counsellor name. The entries are as follows:

ID	Category	Description	Status	Counsellor
2	Fix_Meeting	I need a new meeting	APPROVED	counsellor1
3	Emergency_Support	I AM NEW USER	APPROVED	counsellor1
6	Others	hello this is other from new user	PENDING	counsellor1
8	Fix_Meeting	I would like to have a meeting regarding my health insurance	APPROVED	counsellor1

At the bottom of the table, there is a pagination bar that says "Showing 1 to 5 of 5 entries". To the right of this bar are three buttons: "Previous", "1" (which is highlighted), and "Next".

Figure 31 Counsellor Dashboard

The counsellor can then see the assigned requests to him/her.

5.6 Chat Module

This is one of the most important functionalities of the project, this enables the user to have a text, audio, and video chat with the assigned counsellor. This is the most complex part of the entire project as the most complex issue here is that Django or any native backend framework does not handle WebRTC programming. Audio and Video calls are sent and communicated over WEBRTC protocol and it is a very complex operation.

5.6.1 WebRTC

It is an open-source project or protocol which enables the secure and encrypted transfer of audio, video, and even textual data between peers (WebRTC, 2020). The webRTC in turns uses a TURN/STUN server to establish a connection between peers. The peers cannot directly connect, so the TURN server acts as a central point so that all the peers can connect.

WebRTC Connection Diagram

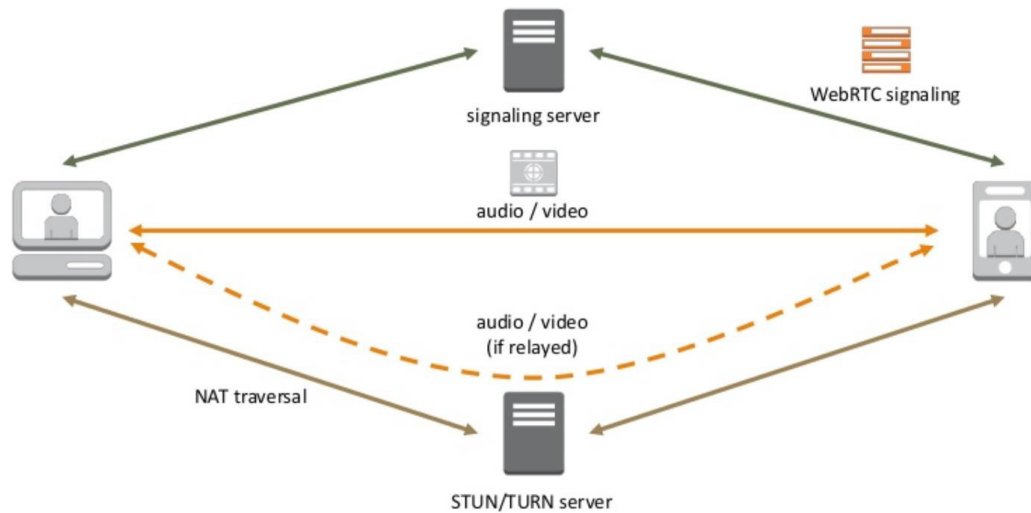


Figure 32 WebRTC diagram

The code for the chatting module is quite complex and most of the webRTC code was written in JavaScript since that is the language that web browsers use. I used a simple Heroku server for the connections and a snippet from the code is pasted below.

```
peerapp = (function() {  
    'use strict';  
  
    console.log("Peer client started");  
  
    var PEER_SERVER = 'my-peer.herokuapp.com';  
    var PORT = 443;  
    var connectedPeers = {};  
    var myPeerID = generateRandomID(4);  
    var peer;  
    var peerIdAlreadyTakenCount = 3;  
    navigator.getUserMedia = navigator.getUserMedia ||  
    navigator.webkitGetUserMedia || navigator.mozGetUserMedia;
```

At the backend, there are also many different views written that supply data to the webRTC server to register peers or users on a call. The user data is saved in the database for Django so there has to be a proper way to transmit this data to the webRTC servers so that a call between the peers could be established.

The custom APIs provided just this in a very effective manner, there are multiple security checks to ensure that integrity is maintained while connecting to the servers and that call information remains safe and secure as well.

```

class ThreadView(View):
    template_name = 'chat/chat.html'

    def get_queryset(self):
        return Thread.objects.by_user(self.request.user)

    def get_object(self):
        other_username = self.kwargs.get("username")
        print("_____")
        self.other_user = get_user_model().objects.get(username=other_username)
        obj = Thread.objects.get_or_create_personal_thread(self.request.user, self.other_user)
        if obj == None:
            raise Http404
        return obj

    def get_context_data(self, **kwargs):
        context = {}
        context['me'] = self.request.user
        context['thread'] = self.get_object()
        context['user'] = self.other_user
        context['messages'] = self.get_object().message_set.all()

        current_logged_user = User.objects.get(username=self.request.user.username)
        group_type = current_logged_user.groups.all()[0].name
        print(group_type)

        if group_type == 'CONSELLOR' or group_type == "G_ADMIN":

```

Figure 33 Class-based view for chat threads.

This picture handles the chat information and generates a thread view for the users. Text-based chatting is handled through 'Django channels', it is a package that encrypts and decrypts chat in real-time while maintaining a good benchmark for speed. The whole application inside the project called 'chat' has many different files corresponding to the chat code. The front end is very simplistic the user can sign in and just see the assigned counsellors, click on the required counsellor, and start the chat. The same case is with the counsellor as well. When a chat is started both the actors can start an audio and video call at the click of a button.

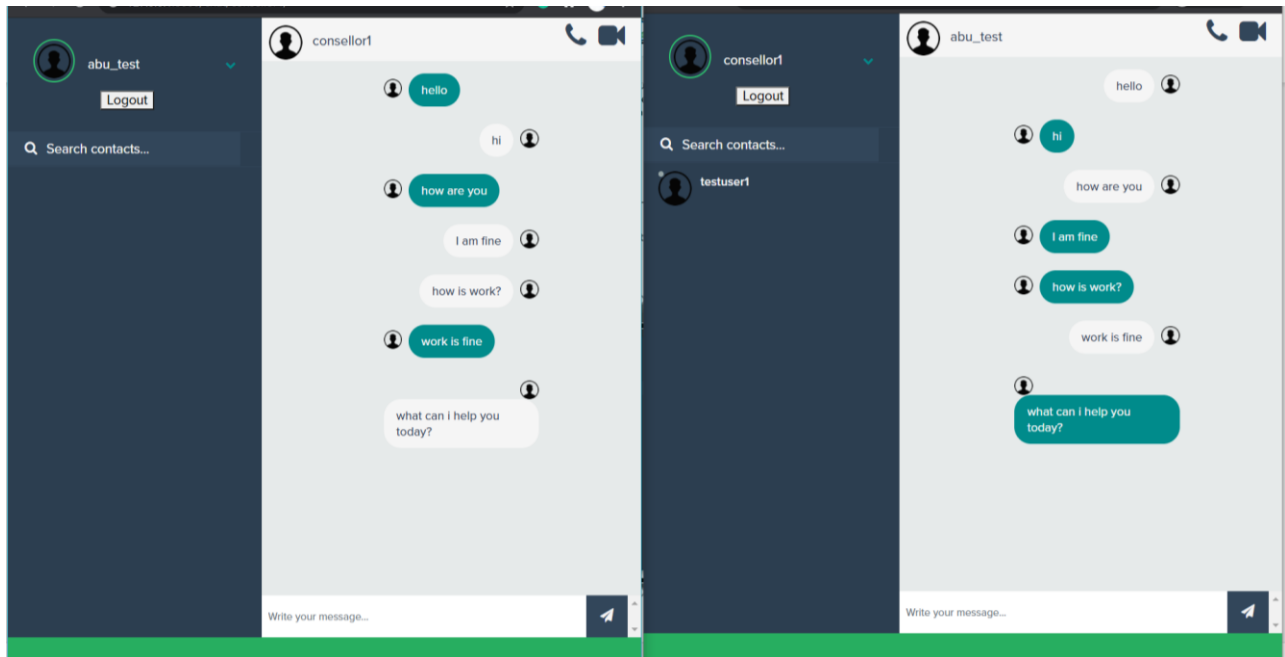


Figure 34 text chat between a migrant and counsellor.

6 Evaluation and Results

6.1 Test Cases

The system contains many different types of users and every user has different types of privileges as well. The need of proper testing is very crucial for a system like this because any system that deals with realistic information and especially legal information needs to be tested properly so that the functionality, as well as security of the system, is benchmarked, and the software goes through a proper testing cycle ensuring the quality of the system according to standards. This system was testing compactly against the requirements and every functionality was counter endorsed by a specific test case. A test case is a documentation artefact recording the steps and procedures of system testing and providing feedback to the result of the test of a module or functionality (Software Testing Help, 2021). A normal test case contains the following:

1. Description
2. Input
3. Expected Response
4. Actual Response
5. Status

Figure 9 shows an example test case for the Login Module. The remaining test cases can be found in the Appendix file "Test Cases".

Test Case ID: TC1			Test Designed by: Abubakar Omer			
Test Priority (Low/Medium/High): High			Test Designed date: 5/23/2021			
Module Name: Login Module			Test Executed by: Abubakar Omer			
Test Title: Verify login with valid username and password			Test Execution date: 5/23/2021			
Description: Test the login page with valid credentials						
Pre-conditions: User has a valid username and password						
Dependencies: None						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to login page	User= example@gmail.com	User should be able to login	User is navigated to dashboard with successful login	Pass	
2	Provide valid username	Password: 1234				
3	Provide valid password					
4	Click on Login button					

Table 6 Example Test Case

Figure 35

Table *n* details all the tests and their results

Test ID	Description	Priority	Pass/Fail
TC1	Test the login page with valid credentials	High	Pass
TC2	Test the login page with invalid credentials	High	Pass
TC3	Test the sign-up page with valid credentials	High	Pass
TC4	Test the sign-up page with invalid credentials	High	Pass
TC5	Test the sign up page with existing username	High	Pass
TC6	Test the password reset functionality for users	High	Pass
TC7	Verify the content is translated to the chosen language	High	Pass
TC8	Test the website content is all viewable	High	Pass
TC9	Test the query creation module	High	Pass
TC10	Test the query creation module	High	Pass
TC11	The user is signed in and a query is created	High	Pass
TC12	Test that the user can see all queries which are only generated by him/her	High	Pass
TC13	Test that the admin can reject the quires generated by the users	High	Pass
TC14	Test that the admin can approve the quires generated by the users	High	Pass
TC15	Test that the admin can assign counsellors to the approved requests available	High	Pass

TC16	Test that the admin can upload new local services		
TC17	Test that the admin can update existing local services		
TC18	Test that the admin can delete existing local services		
TC19	Test that the admin can upload new social services		
TC20	Test that the admin can update existing social services		
TC21	Test that the admin can delete existing social services		
TC22	Test that the counsellor can view all the assigned queries		
TC23	Test that the counsellor cannot edit assigned queries		
TC24	Test that the counsellor can only view users in chat that are assigned to him/her.		
TC25	Test that the counsellor can send messages to available users		
TC26	Test that the counsellor can voice call available, users		
TC27	Test that the counsellor can video call available, users		

Table 7 System Test Table

7 Future Work

The system which is currently deployed serves as a building block for future upgrades and updates. Many of the modules in the system which are currently working have room for improvement. It could be said that software development is an evolutionary process. The cycle of evolution for this project does not end with the current version in use, there will be many more versions of this system. The main targets for the future are focused on many different areas such as improving the stability and the scale of the database structure. The current database schema does indeed work as per the given scenario but in the future, when requirements are more polished and refined according to the user demands, the current schema may cause issues for maintainability. The coding style used in the system can also be improved by using proper design pattern practices, which would make the code more reusable and compact for upgrading. There is also room for improvement in the system for redesigning. Establishing a proper frontend framework can be used to make the front end more flawless and easy to navigate.

The key focus for the future should be in investing time and resources to raise awareness of this system for the targeted audience. This could be achieved by conducting awareness camps and running social media campaigns as well. Once the targeted audience becomes aware of such a facility, the users for the system would increase exponentially. The surveys should also be redesigned and conducted again with a bigger audience. The greater part of the audience expresses their concerns and problems, the more our system would mature. The biggest challenge would be to enrol volunteers to serve as counsellors for the targeted community, who would be willing to provide free help and information to the end-users. A programme to induct volunteer councillors also needs to be launched, such that the system has enough helpers to cater for the needs and requests of our current system. There are multiple horizons this project can achieve with the right sense of leadership and direction, so that more accurate results could be achieved and thus the main issues could be resolved.

8 Conclusion

There is a gap between the migrant community and the information relevant to them. There are different causes for this gap but with the help of the conducted survey, it became clear that the major issue is the

language barrier. The communities find it hard to understand the rules as well as getting basic information regarding healthcare, banking & legal affairs. The COVID'19 pandemic further raised issues for our targeted community as they were unable to physically reach out to the authorities and even had problems understanding the standard operating procedures. The lockdowns and revised rules for medical services made it much harder for the refugees and the migrants to function properly. The existing systems for refugee facilitation lack some of the most major requirements for our community. The system developed serves as a solution to many of the problems faced by the targeted community as it can be used in multiple departments by working professionals. The whole idea of this project focuses on making things easier for the people who had to leave their own countries due to some problems. The system is a step forward in the formation of a diverse society that would be more inclusive for people with different cultural backgrounds.

This system would help to reduce the load on the government departments as well, in many cases, the information needed by the community member is easily available out there but due to lack of awareness regarding it, government officials are engaged. The system helps in providing the same information in multiple languages ready to be shared with anyone and if someone has a more custom request or query, he/she can easily request a chat with a concerned counsellor to get guidance regarding the encountered problem. The usage of this system would benefit society as a whole because it helps in making the newcomers in the UK more responsible citizens. It should be remembered that the single unit of a society is a single person and when one person becomes facilitated, it means as if the entire society has been facilitated.

9 Reflection

This project has taught me many things and the development of such a system is personally related to me and the path I have gone through as well. The problems that have been highlighted in this project are those faced by many people of the migrant community. The beginning of the project was an idea that grew up and evolved into a reality. The concepts of software engineering learnt in the classrooms during my degree were realized through and used as best in the development of this project. This project did not become a reality overnight, rather it took many weeks to just finalize the intimal requirements. This project made me invest more time in learning better techniques for requirements gathering, requirement elicitation and then requirement documentation. The concept of conducting surveys was foreign to me, but I managed to create a questionnaire and compile the results into something fruitful.

The project led me to explore different programming languages and software architectures. While researching the selection of the right framework I discovered the fundamental reasons for diverse programming languages and different techniques. The selection of Django was a very sound decision and while working on it, the rationale becomes stronger as Django turned out to be a very powerful tool enabling fast and compact development. I had not used GitHub to a proper extent before this project and after understanding the whole concept of code retention and tracking changes, I would gladly use it for

all the projects I further develop in the future. Many of the myths regarding software development were shattered and the project taught me that software development is a very creative process, and one has to think outside of the box to accomplish something.

This project has had a very positive impact on my professional as well as personal attitude. I have become more knowledgeable about my subject and have understood that the more one learns about development, there is always so much more that can be discovered and there truly is no limit to innovation. I hope to use the skills achieved and the lessons learnt positively throughout my life.

10 References

- Alexsoft, 2018. *Functional and Nonfunctional Requirements*. [Online]
Available at: <https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/>
[Accessed 8 March 2021].
- Equality and Human Rights Commission , 2018. *Asylum seekers in Britain unable to access healthcare*. [Online]
Available at: equalityhumanrights.com/en/our-work/news/asylum-seekers-britain-unable-access-healthcare
[Accessed 3 March 2021].
- Lydia McMullan, P. D. G. B. P. G., 2021. *COVID Chaos*. [Online]
Available at: <https://www.theguardian.com/world/ng-interactive/2020/dec/16/covid-chaos-a-timeline-of-the-uks-handling-of-the-coronavirus-crisis>
[Accessed 2 March 2021].
- Refugee Cardiff, 2021. *Refugee Cardiff*. [Online]
Available at: <https://www.refugeecardiff.com/>
[Accessed 5 April 2021].
- Software Testing Help, 2021. *How To Write Test Cases*. [Online]
Available at: <https://www.softwaretestinghelp.com/how-to-write-effective-test-cases-test-cases-procedures-and-definitions/>
[Accessed 5 April 2021].
- UK GOV, 2021. *Guidance (COVID'19)*. [Online]
Available at: <https://www.gov.uk/guidance/covid-19-coronavirus-restrictions-what-you-can-and-cannot-do#going-to-work>
[Accessed 21 March 2021].
- UNHCR, 2019. *Global Trends - Forced displacements* , New York: United Nations.
- Visual Paradim, 2020. *What is Sequence Diagram?*. [Online]
Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>
[Accessed 16 April 2021].
- WebRTC, 2020. *WebRTC*. [Online]
Available at: <https://webrtc.org/>
[Accessed 3 May 2021].
- Welsh Government, 2021. *Sanctuary*. [Online]
Available at: <https://sanctuary.gov.wales/>
[Accessed 5 April 2021].

Welsh Refugee Council, 2021. *Welsh Refugee Council*. [Online]
 Available at: <https://wrc.wales/>
 [Accessed 1 April 2021].

11 Appendices

11.1 test cases

Test Case ID: TC2	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Login Module	Test Executed by: Abubakar Omer
Test Title: Verify login with invalid username and password	Test Execution date: 5/23/2021
Description: Test the login page with invalid credentials	
Pre-conditions: User has a valid username and password	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to login page	User= example@3gmail.com	User should not be able to login	User is not able to login and error message is given. Dashboard with successful login	Pass	
2	Provide invalid username	Password: *&(hjgd2e3q				
3	Provide invalid password					
4	Click on Login button					

Test Case ID: TC3	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Sign Up Module	Test Executed by: Abubakar Omer
Test Title: Verify sign up with valid fields	Test Execution date: 5/23/2021
Description: Test the sign up page with valid fields	
Pre-conditions: User was not previously registered on the system	
Dependencies: None	

			Expected Result	Actual Result	Status (Pass/Fail)
Step	Test Steps	Test Data			
1	Navigate to sign up page	User= newuser@gmail.com	User should be signed up in the system	User is signed up and is navigated to the login page to login.	Pass
2	Provide valid username	Password: Testpass123			
3	Provide valid password	Number: 430555971			
4	Provide valid Phone number				
5	Click on submit button				

Test Case ID: TC4	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Sign Up Module	Test Executed by: Abubakar Omer
Test Title: Verify sign up with invalid fields	Test Execution date: 5/23/2021
Description: Test the sign up page with invalid fields	
Pre-conditions: User was not previously registered on the system	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to sign up page	User= newuser@gmail.com	User should be not signed up in the system	User is not signed up and validation error is thrown	Pass	
2	Provide valid username	Password: ""				
3	Provide valid password	Number: 430555971				
4	Provide valid Phone number					
5	Click on Login button					

Test Case ID: TC5	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Sign Up Module	Test Executed by: Abubakar Omer
Test Title: Verify sign up with existing username	Test Execution date: 5/23/2021
Description: Test the sign up page with existing username	
Pre-conditions: User was previously registered on the system	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to sign up page	User= test@gmail.com	User should be not signed up in the system	User is not signed up and 'username already exists' error is displayed	Pass	
2	Provide existing username	Password: Testpass123				
3	Provide valid password	Number: 430555971				
4	Provide valid Phone number					
5	Click on Login button					

Test Case ID: TC6	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Password Reset Module	Test Executed by: Abubakar Omer
Test Title: Verify the password reset module	Test Execution date: 5/23/2021
Description: Test the password reset functionality for users	
Pre-conditions: User was previously registered on the system	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to sign in page	User= test@gmail.com	User should receive URL to update the password and it should be changed	User password is successfully updated	Pass	
2	Click on the password reset button	Password: Testpass123				
3	Provide your email					
4	Click on the link received on the email.					
5	Provide new password and click submit					

Test Case ID: TC7	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Content Translation Module	Test Executed by: Abubakar Omer
Test Title: Verify the content is translated	Test Execution date: 5/23/2021
Description: Verify the content is translated to the chosen language	
Pre-conditions: User is on the website	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to home page	"Spanish"	The website content should be translated to the selected language.	The website content is translated to Spanish	Pass	
3	Click on the Language selection button					
4	Select a language to be used.					

Test Case ID: TC8	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Content Preview	Test Executed by: Abubakar Omer
Test Title: Verify the website content	Test Execution date: 5/23/2021
Description: Test the website content is all viewable	
Pre-conditions: User has accessed the website	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to any services page		User should be able to view the entire content of the website.	User is able to view the entire content and there are no missing fields	Pass	
2	View the available content					

Test Case ID: TC8	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Content Preview	Test Executed by: Abubakar Omer
Test Title: Verify the website content	Test Execution date: 5/23/2021
Description: Test the website content is all viewable	
Pre-conditions: User has accessed the website	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to any services page		User should be able to view the entire content of the website.	User is able to view the entire content and there are no missing fields	Pass
2	View the available content				

Test Case ID: TC9	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Query Creation	Test Executed by: Abubakar Omer
Test Title: Verify the query creation	Test Execution date: 5/23/2021
Description: Test the query creation module	
Pre-conditions: User is signed in	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
------	------------	-----------	-----------------	---------------	--------------------	-------

1	Navigate to the query page	Type: Emergency	User should be able successfully create the query.	User is able to create the query	Pass	
2	Fill in the query details in the required fields	Description: "this is a test query"				
3	Click the submit button					

Test Case ID: TC10	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Query Update	Test Executed by: Abubakar Omer
Test Title: Verify the query updating	Test Execution date: 5/23/2021
Description: Test the query creation module	
Pre-conditions: User is signed in and query is created	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to the all queries page	Type: Emergency	User should be able successfully update the query.	User is able to update the query	Pass	
2	Select the query to be updated	Description: "this is a test query updated"				
3	Fill in the new details					
4	Click the update button					

Test Case ID: TC11	Test Designed by: Abubakar Omer
---------------------------	--

Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Query Deletion	Test Executed by: Abubakar Omer
Test Title: Verify the query deletion	Test Execution date: 5/23/2021
Description: Test the query deletion module	
Pre-conditions: User is signed in and query is created	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to the all queries page		User should be able successfully delete the query.	User is able to delete the query	Pass
2	Select the query to be deleted				
3	Click the delete button				

Test Case ID: TC12	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Query Viewing	Test Executed by: Abubakar Omer
Test Title: Verify the query viewing	Test Execution date: 5/23/2021
Description: Test that the user can see all queries which are only generated by him/her	
Pre-conditions: User is signed in and queries is created	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to the all queries page		User should only see the queries created by	User is only able to view the queries by		

2	View the queries		him/her self	him/her self	Pass	

Test Case ID: TC13	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Query Rejection	Test Executed by: Abubakar Omer
Test Title: Verify the query rejection	Test Execution date: 5/23/2021
Description: Test that the admin can reject the quires generated by the users	
Pre-conditions: Admin is signed in and queries is created	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to dashboard	1. Rejected	Admin should be able to reject the query and email should be sent to user that query is rejected	Admin is able to reject the query and email notification is sent to the user.	Pass	
2	Open the queries listing					
3	Pick any query and perform the required action on it					

Test Case ID: TC14	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Query Approval	Test Executed by: Abubakar Omer

Test Title: Verify the query approval	Test Execution date: 5/23/2021
Description: Test that the admin can approve the queries generated by the users	
Pre-conditions: Admin is signed in and queries are created	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to dashboard	1. Approved	Admin should be able to approve the query and email should be sent to user that query is rejected	Admin is able to approve the query and email notification is sent to the user.	Pass	
2	Open the queries listing					
3	Pick any query and perform the required action on it					

Test Case ID: TC15	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Counselor Assignment	Test Executed by: Abubakar Omer
Test Title: Verify the counselor assignment	Test Execution date: 5/23/2021
Description: Test that the admin can assign counsellors to the approved requests available	
Pre-conditions: Admin is signed in and queries are approved	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to dashboard	1. Counsellor 1	Admin should be able to assign a user request to a counsellor and email should be sent to user that query is assigned to a	Admin is able to assign the request to a counsellor and email notification is also sent.	Pass	
2	Open the queries listing					
3	Pick any approved query and perform					

	the required action on it		professional			

Test Case ID: TC16	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Creating Local Services	Test Executed by: Abubakar Omer
Test Title: Verify the uploading of local services	Test Execution date: 5/23/2021
Description: Test that the admin can upload new local services	
Pre-conditions: Admin is signed in	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to dashboard	1. Name: St. P hospital	Admin should be able to upload the new local service	Admin is able to upload the new services	Pass	
2	Open the local services menu	2. Description: Heart based treatment				
3	Insert the required data	3. Address: XYZ st				
4	Click the submit button					

Test Case ID: TC17	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Updating Local Services	Test Executed by: Abubakar Omer
Test Title: Verify the updating of local services	Test Execution date: 5/23/2021
Description: Test that the admin can update existing local services	
Pre-conditions: Admin is signed in	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to dashboard	1. Name: St. P hospital UPDATED	Admin should be able to update the local service	Admin is able to update the local service	Pass	
2	Open the menu and select the local service to edit	2. Description: Heart based treatment				
3	Insert the required data	3. Address: XYZ st				
4	Click the submit button					
Test Case ID: TC18				Test Designed by: Abubakar Omer		
Test Priority (Low/Medium/High): High				Test Designed date: 5/23/2021		
Module Name: Deleting Local Services				Test Executed by: Abubakar Omer		
Test Title: Verify the deleting of local services				Test Execution date: 5/23/2021		
Description: Test that the admin can delete existing local services						
Pre-conditions: Admin is signed in						
Dependencies: None						

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to dashboard		Admin should be able to delete the local service	Admin is able to delete the local service	Pass	
2	Open the menu and select the local service to edit					
3	Click the Delete button					

Test Case ID: TC19				Test Designed by: Abubakar Omer		
Test Priority (Low/Medium/High): High				Test Designed date: 5/23/2021		

Module Name: Creating Social Services	Test Executed by: Abubakar Omer
Test Title: Verify the uploading of social services	Test Execution date: 5/23/2021
Description: Test that the admin can upload new social services	
Pre-conditions: Admin is signed in	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to dashboard		Admin should be able to upload the new social service	Admin is able to upload the new services	Pass	
2	Open the social services menu					
3	Insert the required data					
4	Click the submit button					

Test Case ID: TC20	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Updating Social Services	Test Executed by: Abubakar Omer
Test Title: Verify the updating of social services	Test Execution date: 5/23/2021
Description: Test that the admin can update existing social services	
Pre-conditions: Admin is signed in	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to dashboard		Admin should be able to update the social	Admin is able to update the social		

2	Open the menu and select the social service to edit		service	service	Pass	
3	Insert the required data					
4	Click the submit button					

Test Case ID: TC21	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Deleting Social Services	Test Executed by: Abubakar Omer
Test Title: Verify the deletion of social services	Test Execution date: 5/23/2021
Description: Test that the admin can delete existing social services	
Pre-conditions: Admin is signed in	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to dashboard		Admin should be able to delete the social service	Admin is able to delete the social service	Pass	
2	Open the menu and select the social service to delete					
3	Click the delete button					

Test Case ID: TC22	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: View Assigned queries – Counsellor Dashboard	Test Executed by: Abubakar Omer

Test Title: Verify the viewing integrity from counsellor's dashboard	Test Execution date: 5/23/2021
Description: Test that the counsellor can view all the assigned queries	
Pre-conditions: Admin is signed in	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to dashboard		Counsellor should be able to view all the assigned queries	Counsellor is able to view all the assigned queries	Pass	
2	Open the menu to view all the assigned queries					

Test Case ID: TC23	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: View Assigned queries – Counsellor Dashboard	Test Executed by: Abubakar Omer
Test Title: Verify the queries can-not be edited from counsellor dashboard	Test Execution date: 5/23/2021
Description: Test that the counsellor can-not edit assigned queries	
Pre-conditions: Admin is signed in	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to dashboard		Counsellor should not be able to edit all the assigned queries	Counsellor is able to edit the assigned queries	Pass	
2	Open the menu to view all the assigned queries					

Test Case ID: TC24	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: View Assigned User – Counsellor Chat	Test Executed by: Abubakar Omer
Test Title: Verify the counsellor can only view users in chat which are assigned to him/her.	Test Execution date: 5/23/2021
Description: Test that the counsellor can only view users in chat which are assigned to him/her.	
Pre-conditions: Counsellor is signed in	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to chat page		Counsellor should only be able to chat with the assigned users	Counsellor is able to chat with only the assigned users	Pass	
2	Open the side bar to see available users					

Test Case ID: TC25	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Chat Module	Test Executed by: Abubakar Omer
Test Title: Verify the counsellor can send messages to users	Test Execution date: 5/23/2021
Description: Test that the counsellor can send messages to available users	
Pre-conditions: Counsellor is signed in	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to chat page	Message: "Hello, this is a test message"	Counsellor should be able to send messages to available users	Counsellor is be able to send messages to available users	Pass	
2	Open the side bar to see available users					
3	Select a user					
4	Write the message and press send					

Test Case ID: TC26	Test Designed by: Abubakar Omer
Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Chat Module	Test Executed by: Abubakar Omer
Test Title: Verify the counsellor can voice call to users	Test Execution date: 5/23/2021
Description: Test that the counsellor can voice call to available users	
Pre-conditions: Counsellor is signed in	
Dependencies: None	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to chat page		Counsellor should be able to voice call to available users	Counsellor is be able to voice to available users	Pass	
2	Open the side bar to see available users					
3	Select a user					
4	press the voice call button					

Test Case ID: TC27	Test Designed by: Abubakar Omer
---------------------------	--

Test Priority (Low/Medium/High): High	Test Designed date: 5/23/2021
Module Name: Chat Module	Test Executed by: Abubakar Omer
Test Title: Verify the counsellor can video call to users	Test Execution date: 5/23/2021
Description: Test that the counsellor can video call to available users	
Pre-conditions: Counsellor is signed in	
Dependencies: None	

			Expected Result	Actual Result	Status (Pass/Fail)
Step	Test Steps	Test Data			
1	Navigate to chat page		Counsellor should be able to video call to available users	Counsellor is be able to video to available users	Pass
2	Open the side bar to see available users				
3	Select a user				
4	press the video call button				