# Problem Statement

A client in the area of palliative care in the rural sector is concerned about the multiple potential contact points for patients and their carers. Examples of these carers could be. family, neighbours or friends. There is a desire for a system to ensure a single point of contact for the patients/carers. This point of contact would act as a triage point, and provide advice, refer the patient to a higher level of advice, or advise the patient to seek external treatment. This system would need to store information about the contact, including who called, who answered the contact, and the advice given or actions taken.

# ERD Diagram

The ERD Diagram can be found in a separate file.

## Design decisions and normalisation

I believe that the **patient** must provide all the information, namely

* full name
* Date of Birth
* full address, including the region, as it is important to know if the patient lives in rural area.
* home phone, because patients are elderly, sometimes they reject technology, it is difficult for them to adapt to technology due to their advanced age.
* mobile phone
* email (who knows, maybe there are qualified IT specialists among the patients)
* notes - for more information

**Carer**

For the **carer** table, I decided that the following data would be needed

* Full name and surname
* Date of Birth
* What is the relationship between patient and carer
* Full address, including region and district
* Home phone
* Mobile phone
* Email
* Notes - for additional information

The **contactor** can be both a patient and a guardian (if you remove all other persons who can call, but I believe that such persons will not be recorded). Table contactor should have a description or note of a call itself and contain carerId and patientId. I think it is very important to specify that Contactor should be patient OR carer, I mean on what behalf caller is calling. Can it be a device? Like patient will have a button on special equipment (I am imagining a hospital bed with a button to call a nurse) to make call, if patient cannot talk for example. It will provide special message automatically which will contain just a patientId.

Table **worker** should contain standard information full name and surname, mobile(work) phone. An additional field that I believe should be present is the worker's availability, as it may not be available. At the moment it's Boolean, but maybe it's worth changing to two fields instead of one, one of them is the date of the available from, the other is available to. Additional information such as address, etc. I consider irrelevant in this situation, as it may be stored elsewhere.

**Contact** table has a contactID as a primary key, and contactorID, patientId, workerId as foreign keys referencing the corresponding tables. It has a date and time when the contact was made, what the reason for calling, actions that were taken and any additional notes if required.

**Bridging** table. I have a feeling that I am missing a bridging table between patient and carer.

# Security

## GDPR

To make a database GDPR compliant, I will have to ensure the following steps are covered/implemented:

* **Create and enforce roles and permissions.** By defining roles and permissions on the database level in advance, I can prevent unauthorized access to sensitive personal data. This is step can assist in preventing catastrophic errors, data breaches, unethe and loss of data
* **Mask sensitive data.** Data masking within the database ensures that the developers and engineers can use real data when working on their databases, without compromising any individual’s privacy or breaching any articles of the GDPR.
* **Prevent data loss and alteration.** Maintaining the integrity of data is just as important as the security measures used to collect and store it. GDPR Article 32(2) refers to the measures that must be taken to prevent “accidental or unlawful destruction, loss, alteration, unauthorized disclosure of or access to personal data.” Putting aside intentional edits, disclosure or deletions made to the data, accidental removal or edits to the data are a bit harder to prevent. I plan on implementing hashing and encryption in the database.

## GDPR Compliance

In Iteration 1, the database was not GDPR compliant as it did not have any security measures implemented.

# Coding

## Testing

### Iteration 1

In Iteration 1, I implemented the following tables:

* Carer
* Patient
* Contactor
* Contact
* Worker

# Backup

A backup of all personal and general data should be made at the beginning, immediately after the information has been received, as unforeseen circumstances can happen at any time.

# Evaluation

The hardest part is always getting started.

Iteration 1 performance was satisfying. It was hard to divide all the work into equal iterations as some things like security or views are hard to do without fully implementing the database.

It was difficult to come up with a relevant field for tables, cutting off everything unnecessary and non-relevant. Since I have no experience specifically in this industry, it is quite possible that I missed some important information. I hope that in the next iteration I will correct my shortcomings.