To create new answers for new illness type, we need to modify the three following tables:

1. Question\_Type – where we store the type of illness (teachers assign tasks based on illness type)
2. ECGDiagram – where we store the path to the image (note that images should be uploaded as well)
3. CheatSheet – where we specify for which question has what answer

Since there is no telling how this app will be maintained in the future we will explain from the perspective of entity diagrams with some examples of how the application itself was preparing data using ORM.

# Entity Diagram

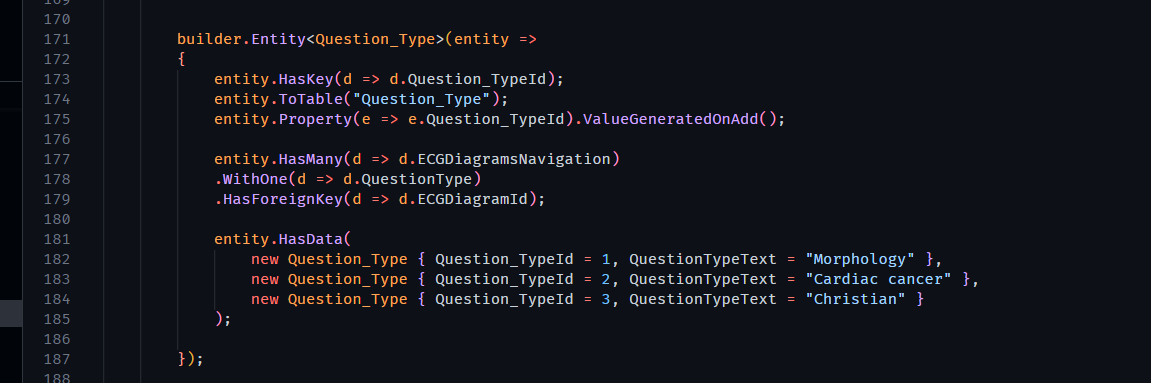
Just for the sake of having a visualization, here is the entity diagram for the database.

Diagram, engineering drawing

Description automatically generated

# Question\_Type

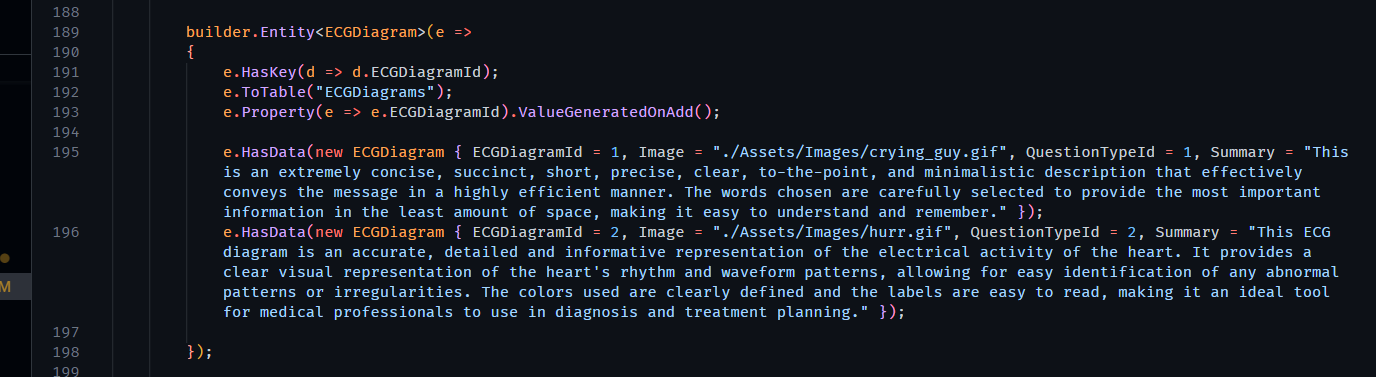
All that is required for Question\_Type table is the QuestionTypeText. It is a simple table used for storing illness names, just make sure you enter proper name.



# ECGDiagram

Within ECGDiagram table, we are storing:

1. ECGId – the primary key
2. QuestionTypeId – the foreign key from Question\_Type table
3. Image - the path to the ECG images that is related to the ECGDiagram
4. Summary (extra information about ECGDiagram/illness on summary page).



# CheatSheet

Before continuing on the CheatSheet table, make sure that you have a record for the previous tables. Inside this table, we are storing:

1. EcgId – the foreign key id of the ECGDiagram (make sure an ECGDiagram has been created already)
2. CheatId – the primary key for the CheatSheet
3. ParentNumber – the question number of the parent question
4. QuestionNumber – the number of the subquestion within the parent question
5. AnswerNumber – the index of an answer within a subquestion
6. Answer – the answer in string

Continue reading first for more context.

EcgId and CheatId are self explanatory so we won’t go deeper.

ParentNumber, QuestionNumber, AnswerNumber and Answer. The reason why we have such columns is mainly because of the format of the questions. Some questions have subquesitons, some questions are single choice question and some are multi-choice question.

1. Most questions have subquestions and some do not. Those questions that do not have subquestions are actually treated as having subquestions as well to keep everything the same.

Imagine the following:

**Questions with subquestions**

Question 1

* Subquestion 1
  + AnswerNumber1
  + AnswerNumber2
  + …
* Subquestion 2
* …

**Questions without subquestions**

Question 1

* AnswerNumber1
* AnswerNumber2
* …

Notice that **questions without subquestion** is essentially questions that have only 1 subquestion.

1. Each subquestion may have different number of possible answers (we use AnswerNumber to index and identify them)
2. Answer in string because we may have answers in the form of decimals.

## How is CheatSheet being used?

There are two types of answers, Boolean and Decimals.

For Boolean we actually only need to know about the True value so we do not store False values at all.

For Decimals, they of course need to be filled in.

In CheatSheet table, we are saying that:

**For AnswerNumber X in the QuestionNumber X of ParentNumberQuestion X is some string value.**

OR

**For Answer X in Subquestion X of ParentQuestion X is some string value.**

## How to add data?

All that is needed is to follow the question format provided by the doctor and simply fill in the answer of a question. And we need to make sure the indexing is correct. (Indexing starts from 1 for every category)

Remember that all questions are treated as having at least ONE subquestion.

