The Case of Deadly Healthcare

Irony Intended

Our Team

Xander Hayhoe

Luca Bastone

Komal Vachhani

Syed Zafar Alam

Table of Contents

The Problem

Solve the ever deteriorating healthcare system in Canada

Currently, the healthcare system struggles with determining when to admit or reject a patient in the ER

Our Solution

Our team solves that problem by removing two key factors:

1. Assessing a patient solely via nurses
2. The accidental admittance of a patient who is fine

The Implementation

Our code is written in **Python** utilising the JSON package to parse data

Python is:

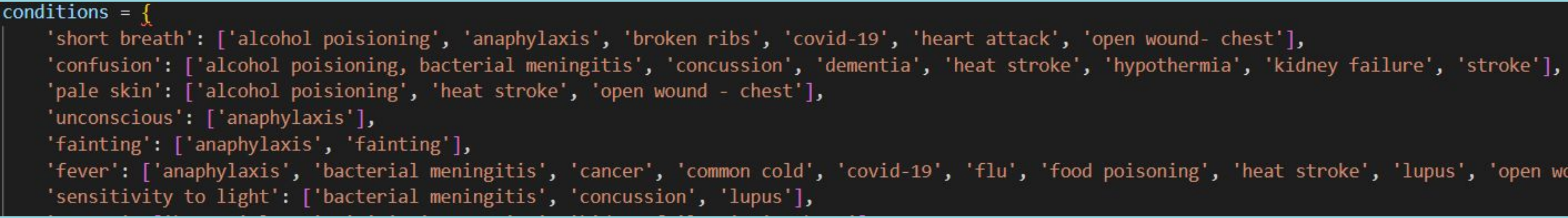
- **Lightweight**
- **Flexible**
- **Maintainable**
- **Scalable**

Django and Flask are some great examples of Full Stack frameworks that allow our project to be easily integrated into an application.

Python and R are the two most powerful tools for managing data.

The Code - Data

To manage our data, we mainly use dictionaries

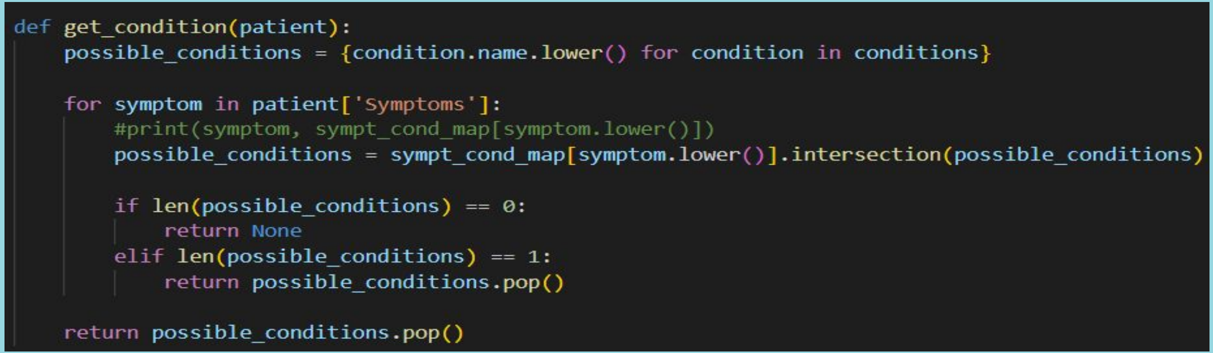


```
conditions = {
    'short breath': ['alcohol poisioning', 'anaphylaxis', 'broken ribs', 'covid-19', 'heart attack', 'open wound- chest'],
    'confusion': ['alcohol poisioning, bacterial meningitis', 'concussion', 'dementia', 'heat stroke', 'hypothermia', 'kidney failure', 'stroke'],
    'pale skin': ['alcohol poisioning', 'heat stroke', 'open wound - chest'],
    'unconscious': ['anaphylaxis'],
    'fainting': ['anaphylaxis', 'fainting'],
    'fever': ['anaphylaxis', 'bacterial meningitis', 'cancer', 'common cold', 'covid-19', 'flu', 'food poisoning', 'heat stroke', 'lupus', 'open wo
    'sensitivity to light': ['bacterial meningitis', 'concussion', 'lupus'],
```

We use symptoms as the keys and store the conditions as the values to boost runtime

The Code - Algorithm

- We use a process of elimination to match a patient's symptoms to the correct conditions



```python
def get_condition(patient):
    possible_conditions = {condition.name.lower() for condition in conditions}

    for symptom in patient['Symptoms']:
        #print(symptom, sympt_cond_map[symptom.lower()])
        possible_conditions = sympt_cond_map[symptom.lower()].intersection(possible_conditions)

        if len(possible_conditions) == 0:
            return None
        elif len(possible_conditions) == 1:
            return possible_conditions.pop()

    return possible_conditions.pop()
```

For each symptom -> match with possible conditions as set

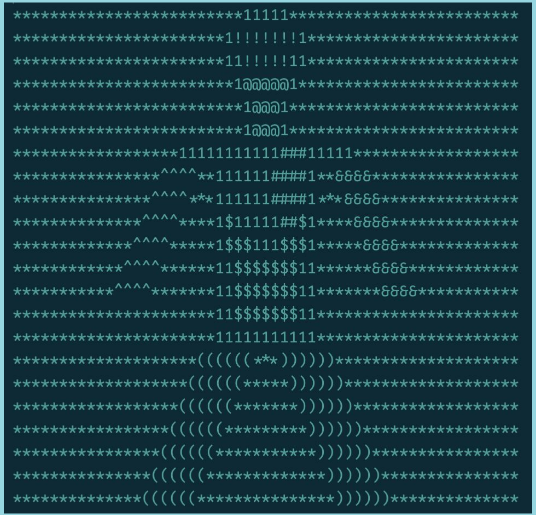
- We calculate intersections of sets to diagnose the patient

The Code - Algorithm cont.

To display the affected areas, we utilize a template string

We use a dictionary to map each body part to a specific character so that it can be replaced in the template string

We fill in the appropriate areas based on the affected areas of the given condition

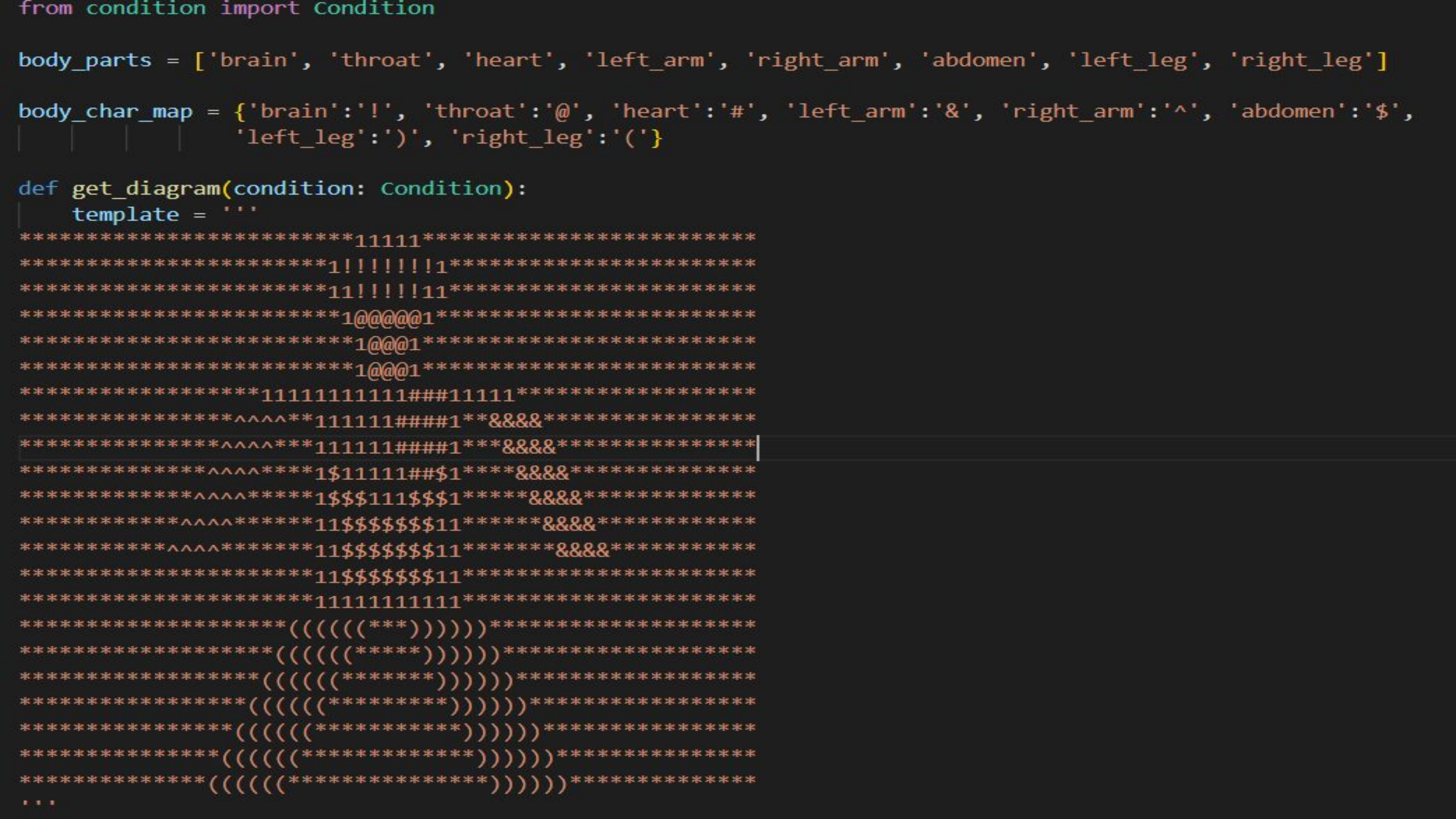


```
*************************11111*************************
***********************1!!!!!!!1***********************
***********************11!!!!!11***********************
************************1@@@@@1************************
*************************1@@@1*************************
*************************1@@@1*************************
******************11111111111###11111******************
****************^^^^**111111####1**&&&&****************
***************^^^^***111111####1***&&&&***************
**************^^^^****1$11111##$1****&&&&**************
*************^^^^*****1$$$111$$$1*****&&&&*************
************^^^^******11$$$$$$$11******&&&&************
***********^^^^*******11$$$$$$$11*******&&&&***********
**********************11$$$$$$$11**********************
**********************11111111111**********************
********************((((((***))))))********************
*******************((((((*****))))))*******************
******************((((((*******))))))******************
*****************((((((*********))))))*****************
****************((((((***********))))))****************
***************((((((*************))))))***************
**************((((((***************))))))**************
```

The Code - Output

-To output our result, we convert the JSON string to a Python string, manipulate it, then convert it back to JSON

-This allows us to perform operations using Python



```python
from condition import Condition

body_parts = ['brain', 'throat', 'heart', 'left_arm', 'right_arm', 'abdomen', 'left_leg', 'right_leg']

body_char_map = {'brain':'!', 'throat':'@', 'heart':'#', 'left_arm':'&', 'right_arm':'^', 'abdomen':'$',
                'left_leg':')', 'right_leg':'('}

def get_diagram(condition: Condition):
    template = '''
*************************11111*************************
***********************1!!!!!!!1***********************
***********************11!!!!!11***********************
************************1@@@@@1************************
*************************1@@@1*************************
*************************1@@@1*************************
******************11111111111###11111******************
****************^^^^**111111####1**&&&&****************
***************^^^^***111111####1***&&&&***************
**************^^^^****1$11111##$1****&&&&**************
*************^^^^*****1$$$111$$$1*****&&&&*************
************^^^^******11$$$$$$$11******&&&&************
***********^^^^*******11$$$$$$$11*******&&&&***********
**********************11$$$$$$$11**********************
**********************11111111111**********************
********************((((((***))))))********************
*******************((((((*****))))))*******************
******************((((((*******))))))******************
*****************((((((*********))))))*****************
****************((((((***********))))))****************
***************((((((*************))))))***************
**************((((((***************))))))**************
'''
```

Real Life Implementation

- Scalability and maintenance is quite simple
- CI/CD pipeline to alter database/algorithm
- Data storage scaled to either cache or database depending on whether we want to keep user data.
- Implement frontend in a website with a ‘waiting line’ and ticket format that automatically assigns the patient their sector
    - Wait time for the sector reduced due to automation

Thank you!

Any questions?