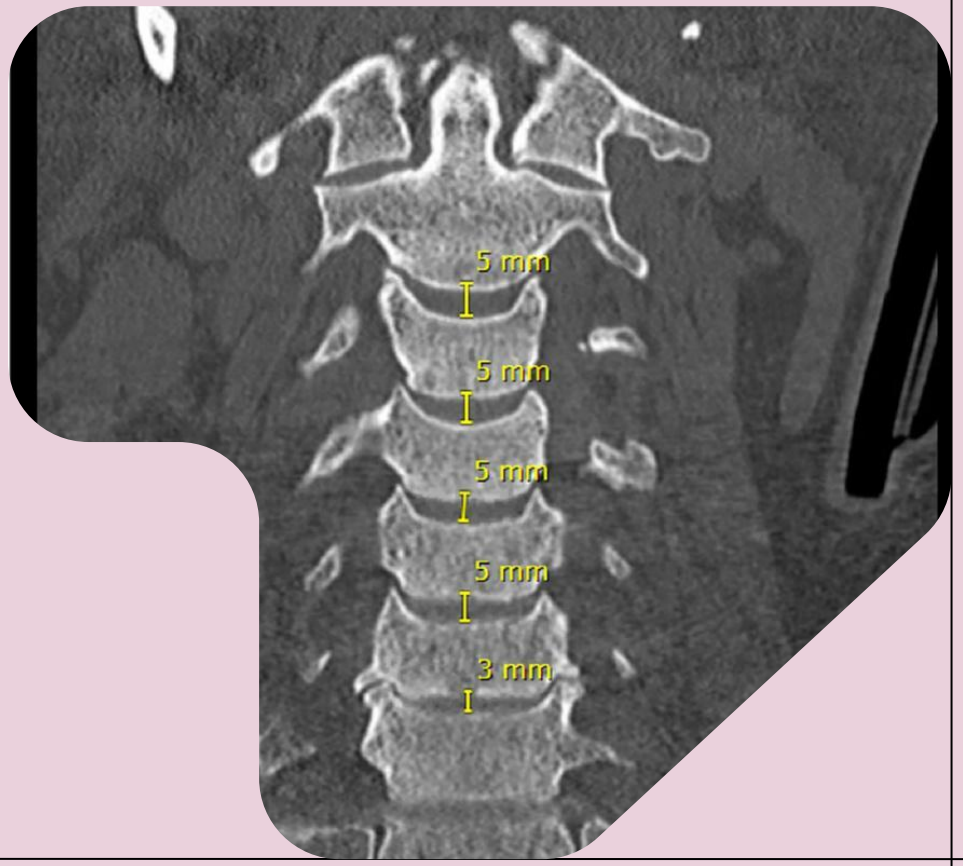
Cervical Spine Fracture Detection



Team FineSpine : Himanshu, Aayush, Saurabh

Problem Definition

- There has been a rise in the incidence of spinal fractures in the elderly and in this population, fractures can be more difficult to detect on imaging due to superimposed degenerative disease and osteoporosis.

- Quickly detecting and determining the location of any vertebral fractures is essential to prevent neurologic deterioration and paralysis after trauma.

Proposed Solution

- Creating an application that assists radiologists with Cervical Spine Fracture detection.
- Improve fracture detection accuracy.
- Reduce reporting time.
- Reduce workload.

Project Scope

Proof Of Concept (POC)

- Work on the RSNA kaggle competition
- Perform EDA
- Experiment on some baseline models.
- Compare models on different metrics. (ROC/AUC/Inference)
- Segmentation and Explainable AI

Prototype

- Create a mockup of screens to see how the app could look like
- Deploy one model to Fast API to service model predictions as an API

Minimum Viable Product (MVP)

- Create an app to detect cervical spine fracture
- API Server for uploading CT scan and predicting and segmenting the fracture.

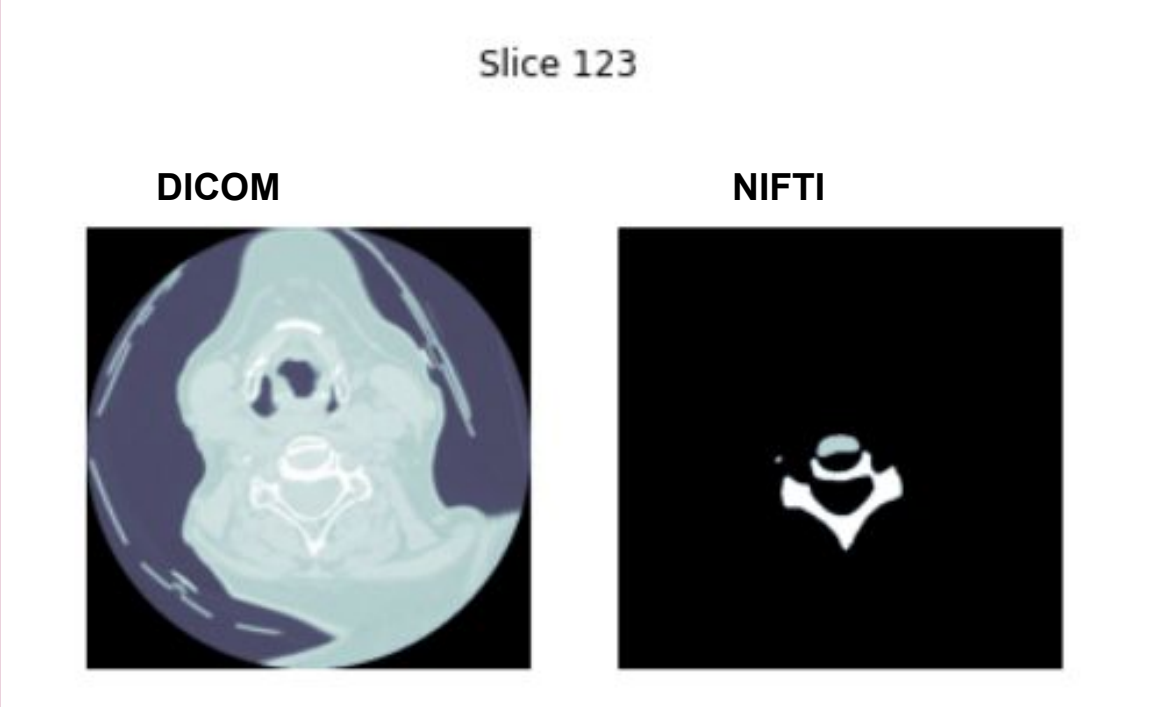
Data

- We are using the data provided by RSNA kaggle challenge
- It has 2019 CT scans(dicom and nii files ~300 GB)
- Label Counts:
  - fractured vertebrae: 1058
  - normal vertebrae: 961

DICOM and NIFTI files

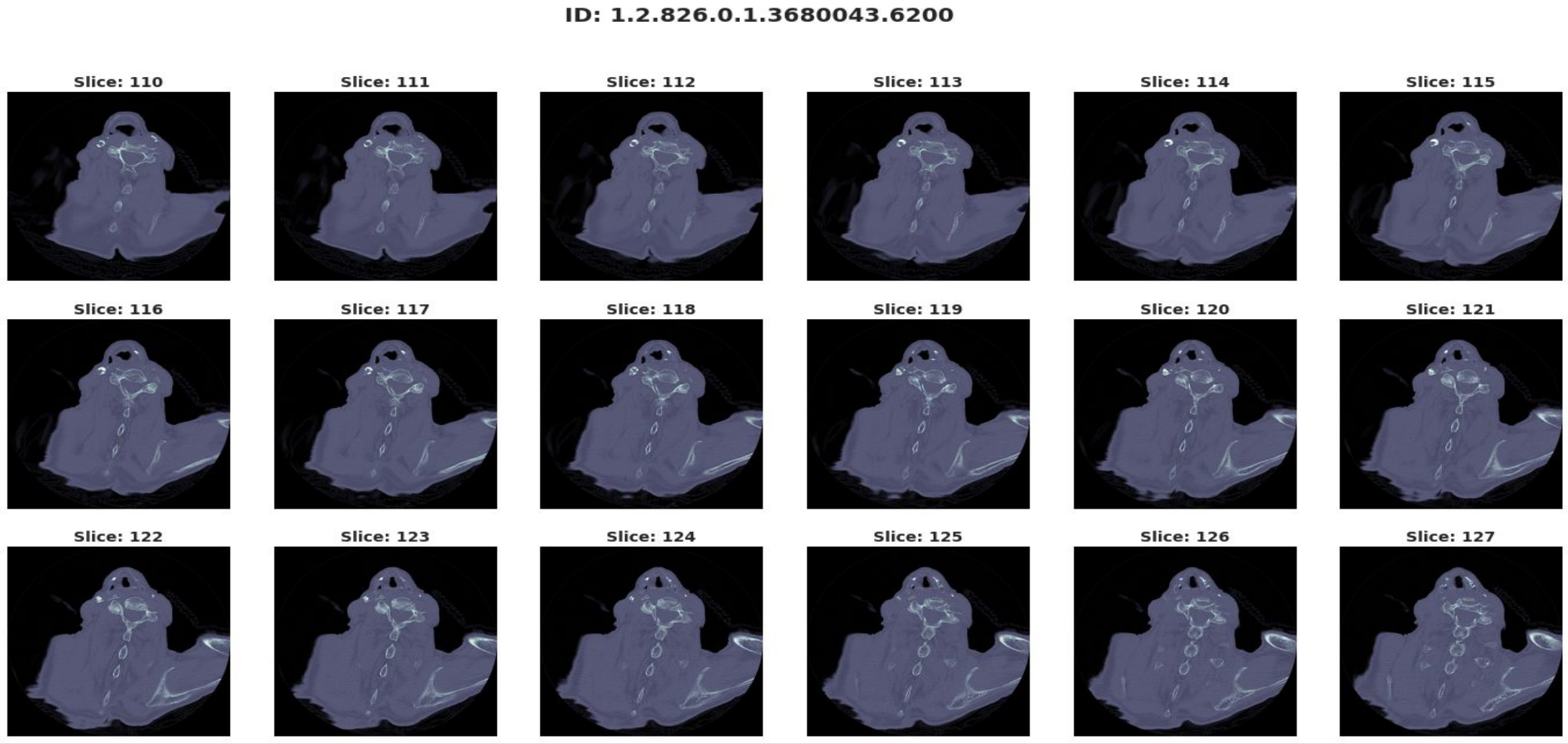
`.dcm` - DICOM file: A DCM file is an image file saved in the Digital Imaging and Communications in Medicine (DICOM) image format. It stores a medical image, such as a CT scan or ultrasound, and may also include patient information to pair the image with the patient.

`.nii` - NIFTI file format - A nibabel image object is the association of three things:an N-D array containing the image data;a (4, 4) affine matrix mapping array coordinates to coordinates in some RAS+ world coordinate space (Coordinate systems and affines);image metadata in the form of a header.

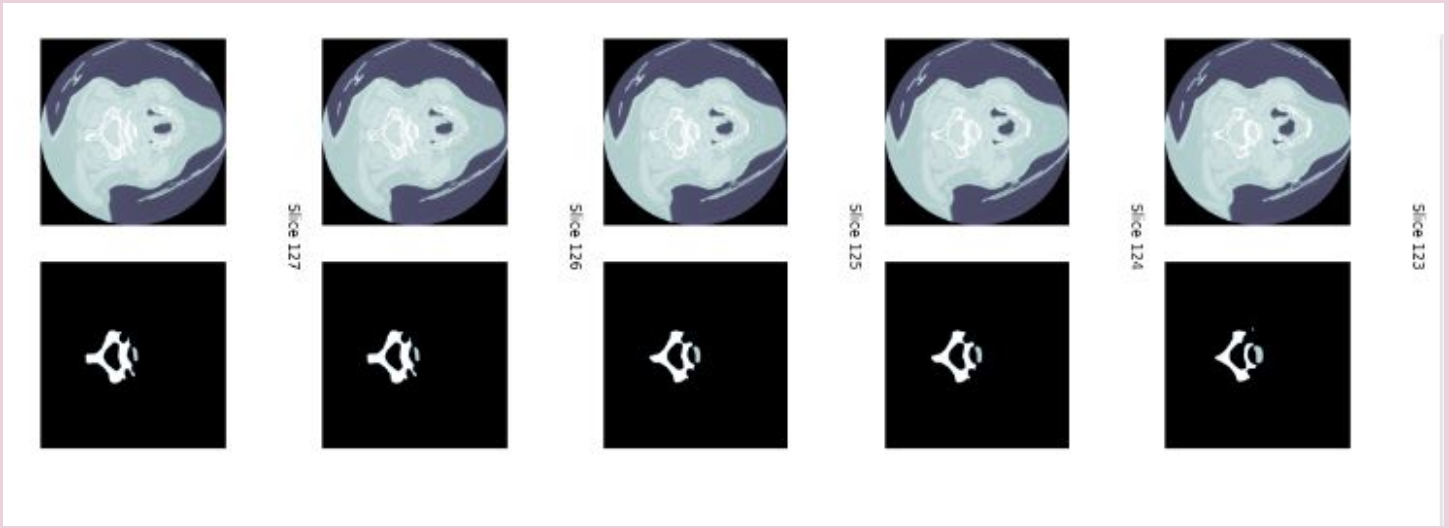


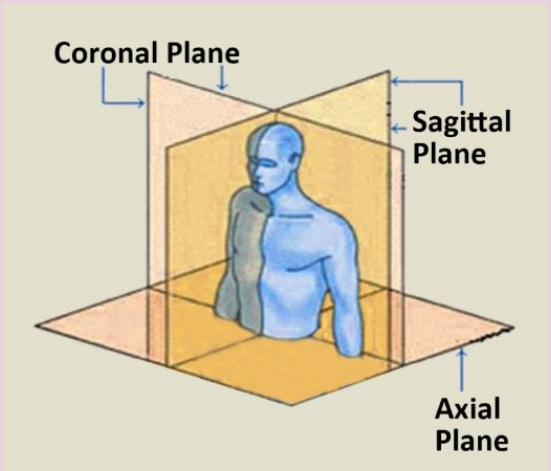
Data

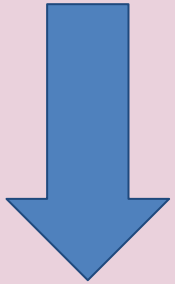
Input data (.dcm and .nii files) are slices of the vertebrae in axial plane.

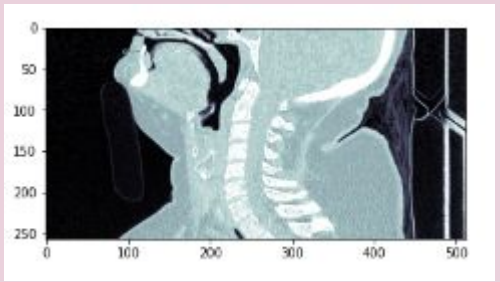


Data Preprocessing



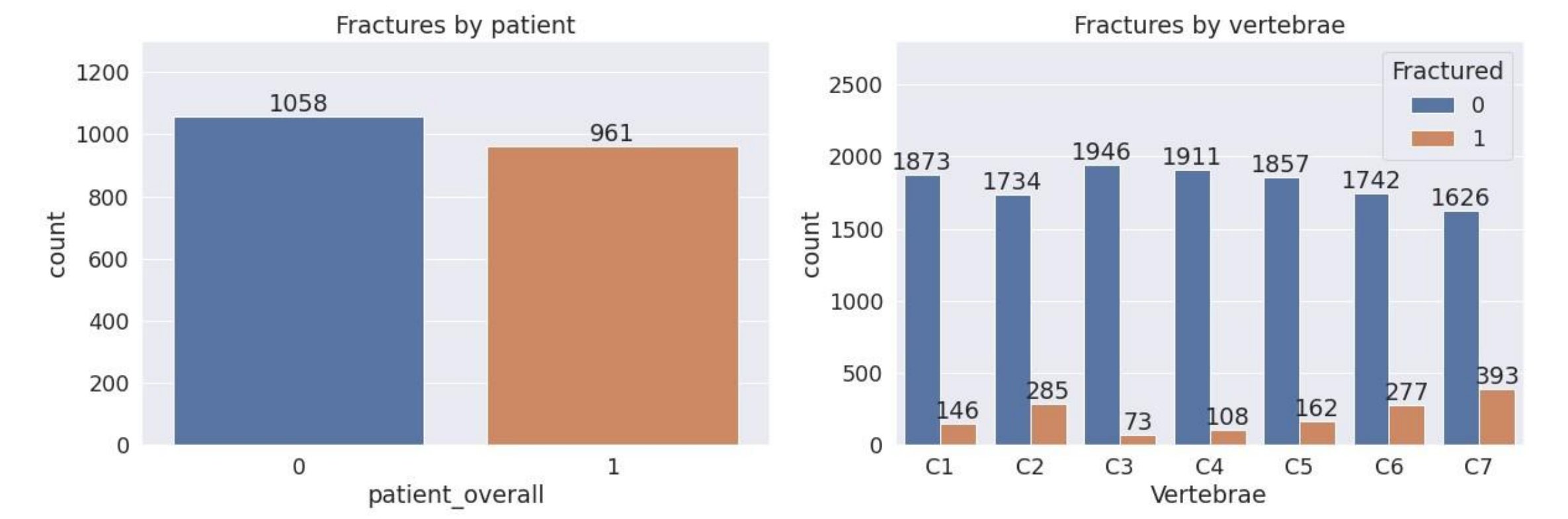






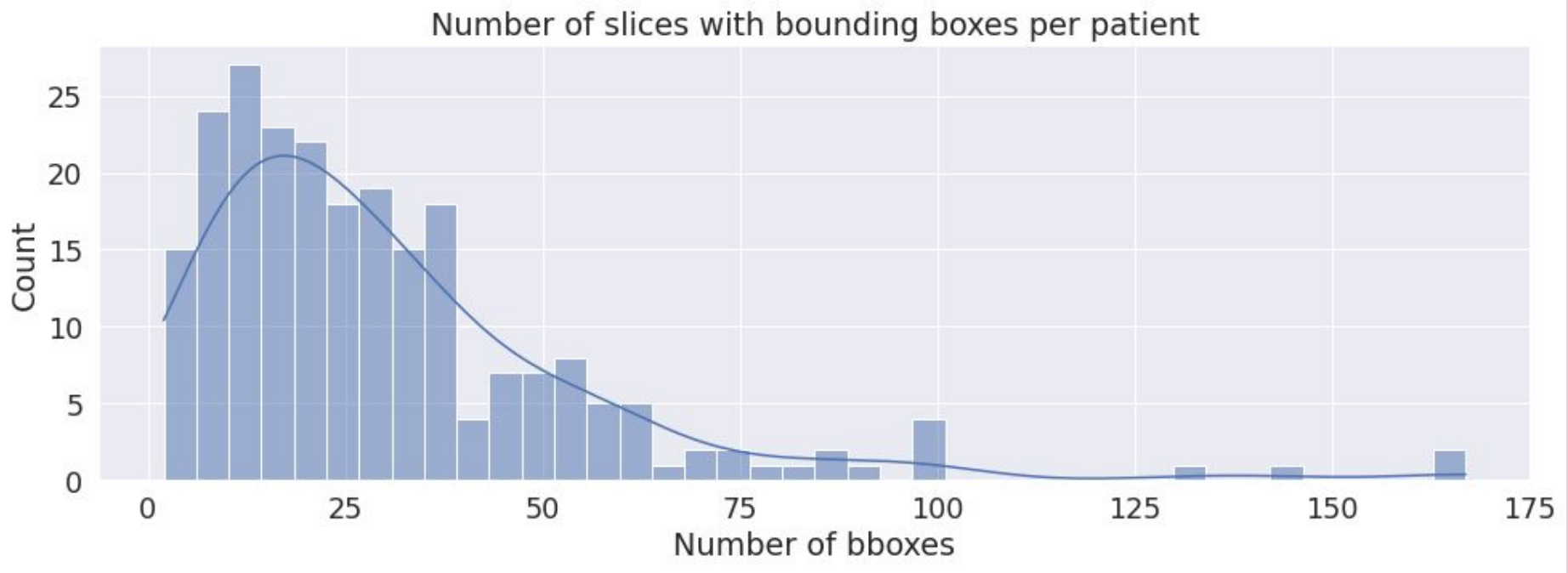
Convert Spinal Cord images from axial plane to Sagittal plane

EDA



- The overall target is roughly balanced (52/48 split).
- C7 has the highest proportion of fractures (19%) whereas C3 has the lowest (4%).

EDA

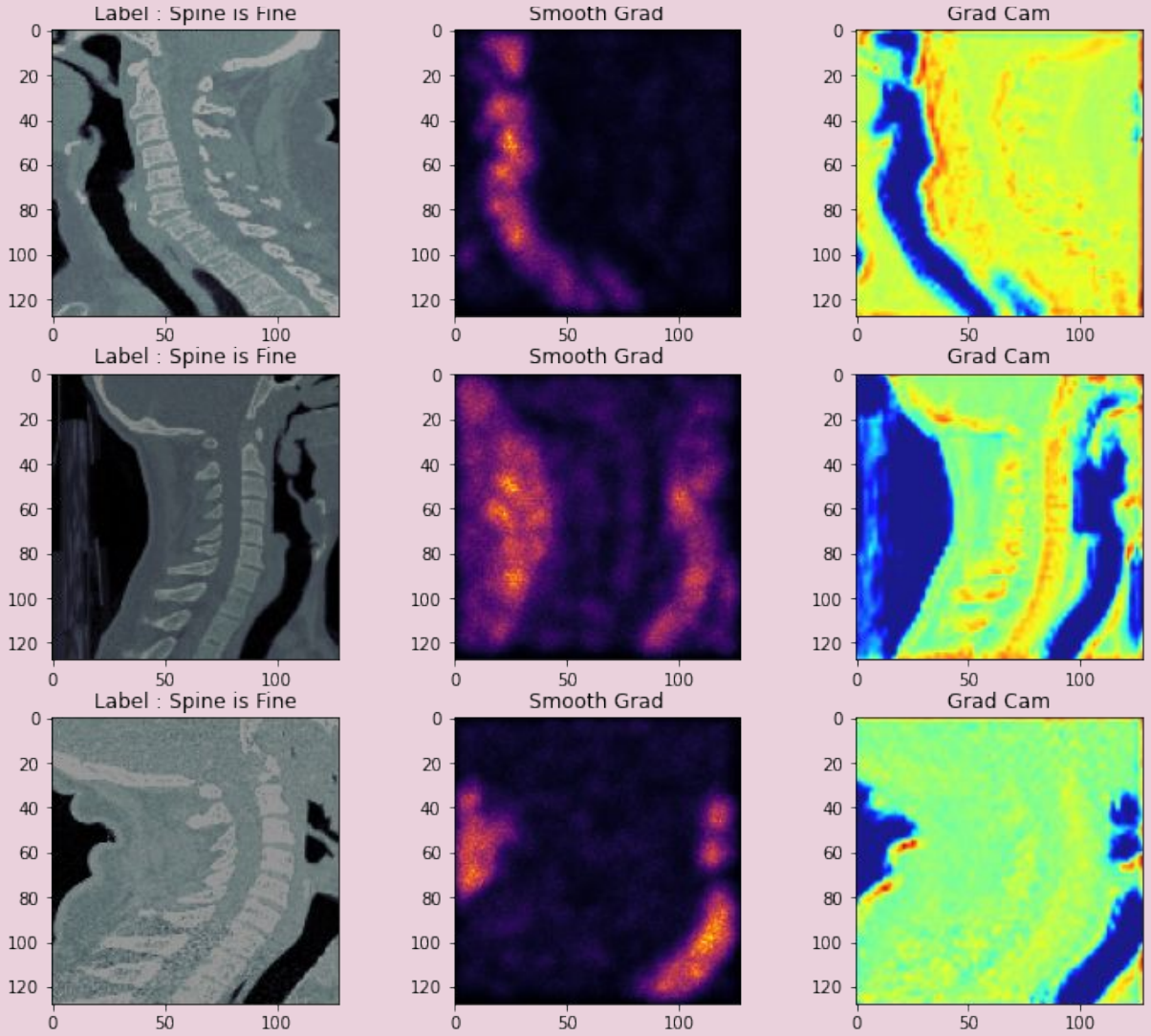


- We are only given bounding boxes for a subset of the data. In particular, only 12% of patients in the train set have any bounding box measurements.
- This information is useful in telling us exactly where the fractures have occured.

Fracture Detection

Sr. No	Model	Validation Accuracy
1.	Exception	60%
2.	Resnet	55%
3.	EfficientNet	52.8%
4.	ConvNet model v1	61%
5.	ConvNet model v2	47.6%
6.	AlexNet	47.6%

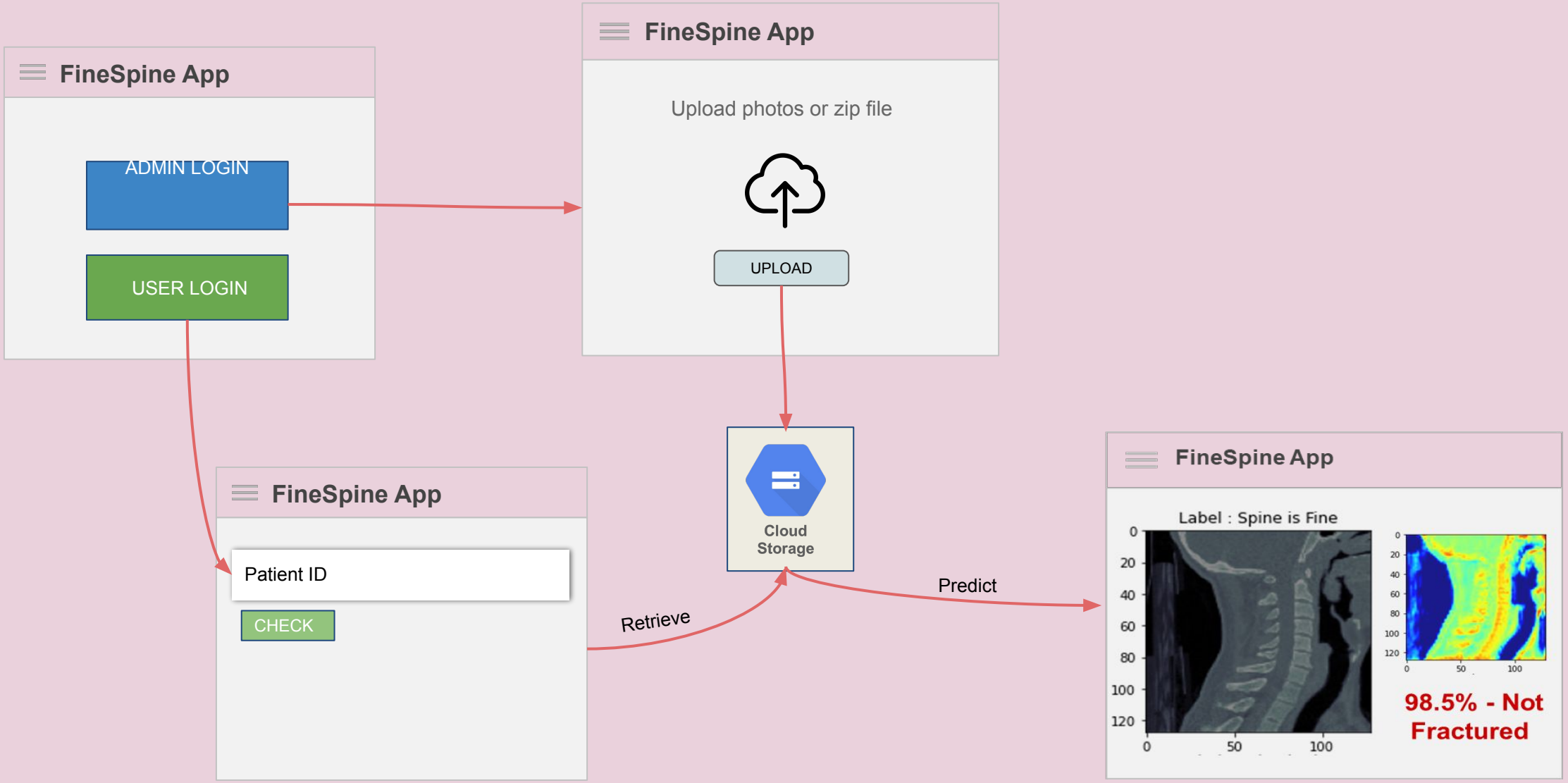
GradCam & Smooth Grad



Fracture Identification

Sr. No	Model	Validation Accuracy
1.	Exception	6%
2.	Resnet	5.2%
3.	EfficientNet	7.6%
4.	ConvNet model v1	13.3%
5.	ConvNet model v2	12.9%
6.	AlexNet	20.5%

Screenflow & Wireframes



Solution Architecture

Process (People)

- Collect data from Kaggle
- Data Processing and EDA
- Model training/tuning
- Upload dicom files
- View prediction and explainability results
- Build App

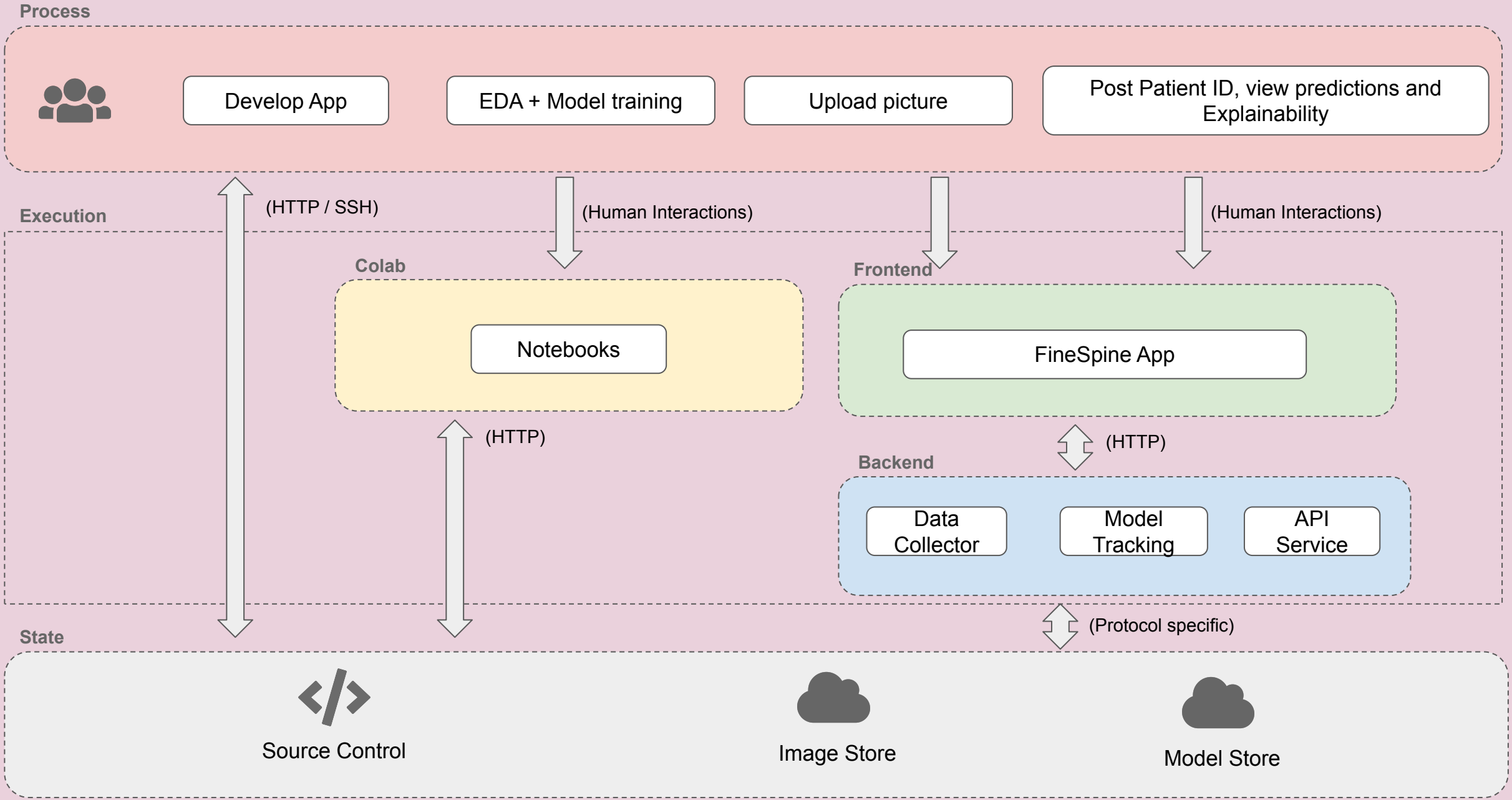
Execution (Code)

- Request customer id and extract customer dicom files from GCP
- Apply same preprocessing steps
- Use the best model to make prediction
- Return results to user
- Spine Fracture or not
- Show model explainability

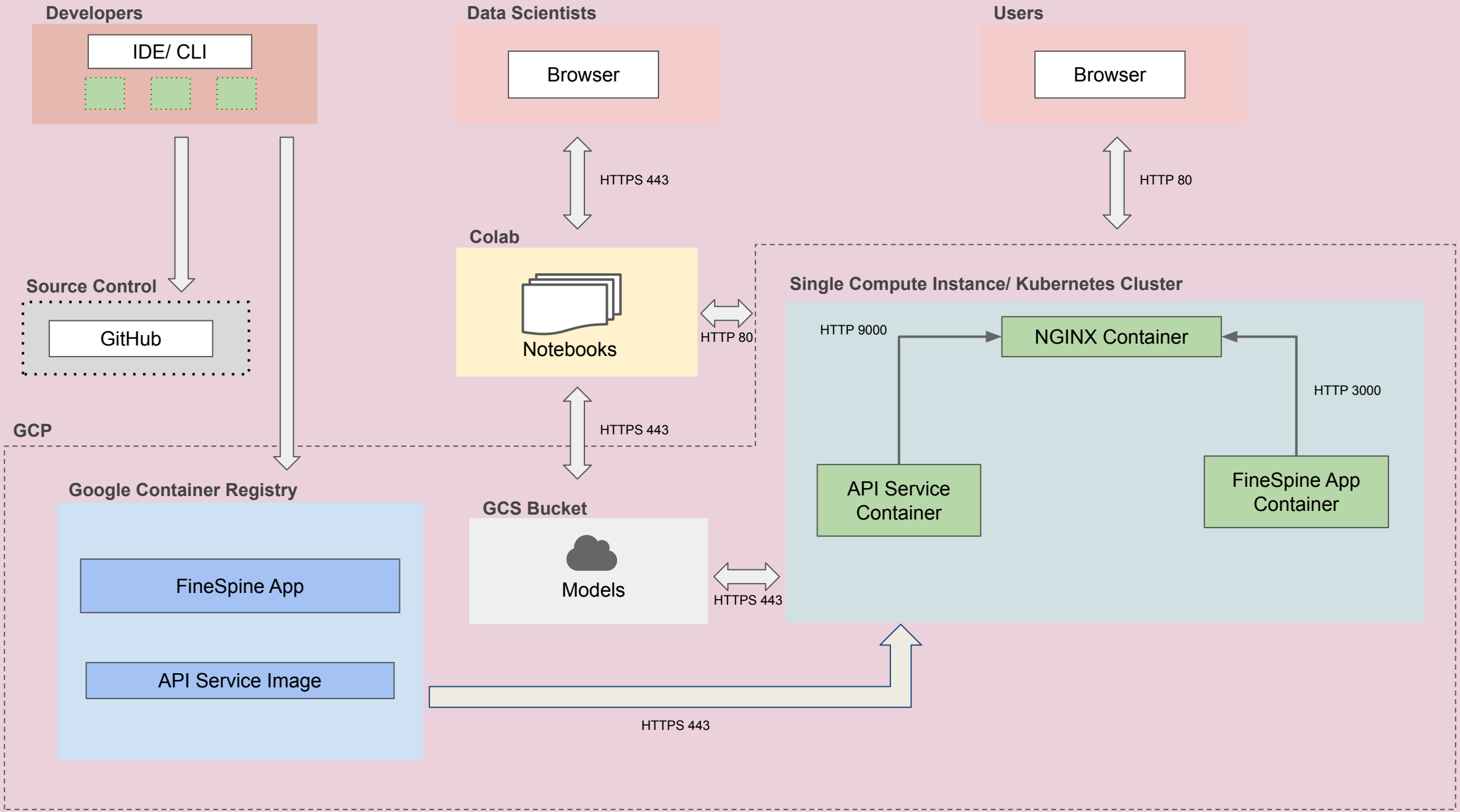
State (Source, Data, Models)

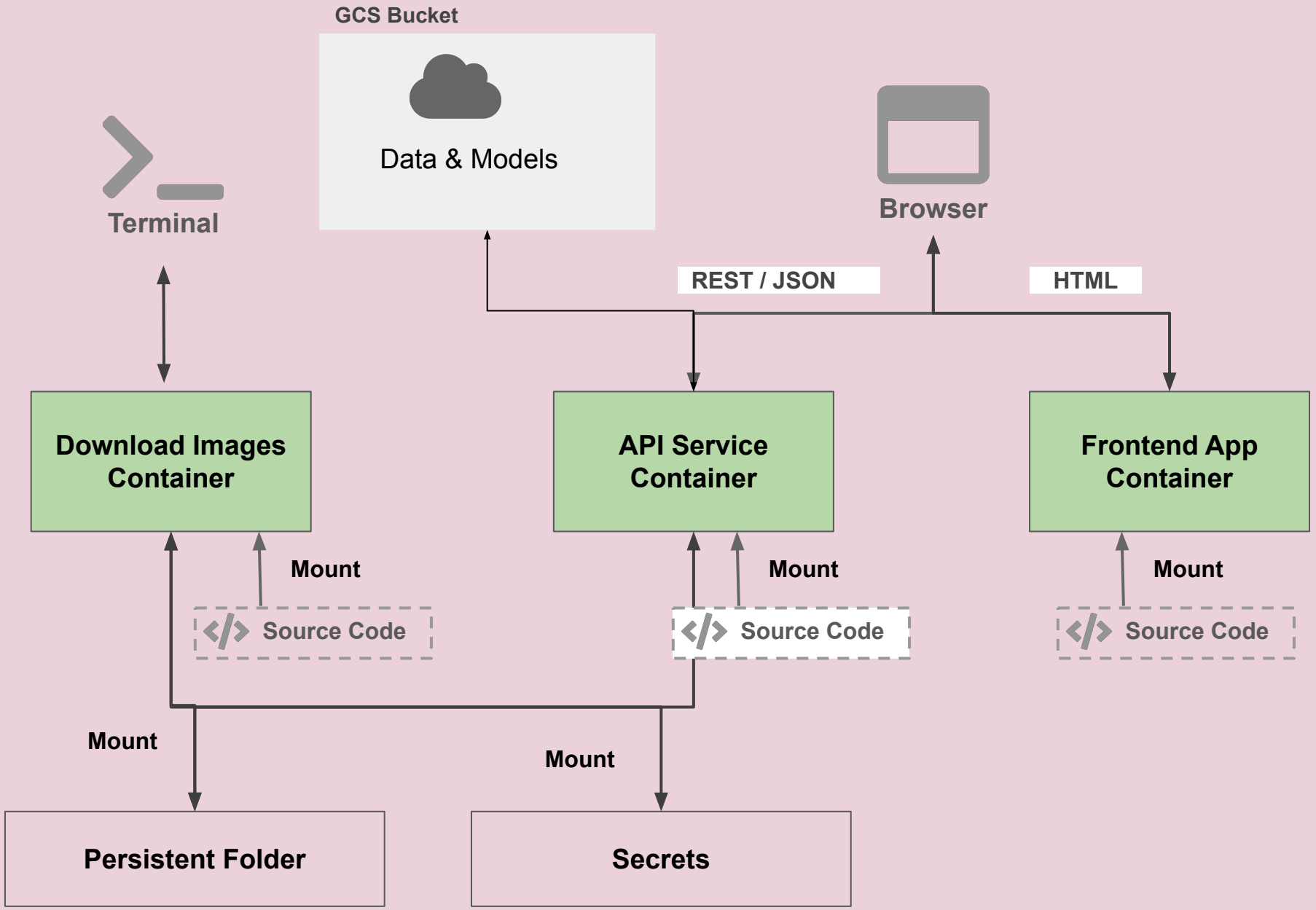
- Save images to GCP bucket
- Save model weights
- Information on preprocessing

Solution Architecture - Group 18 - FineSpine



Technical Architecture

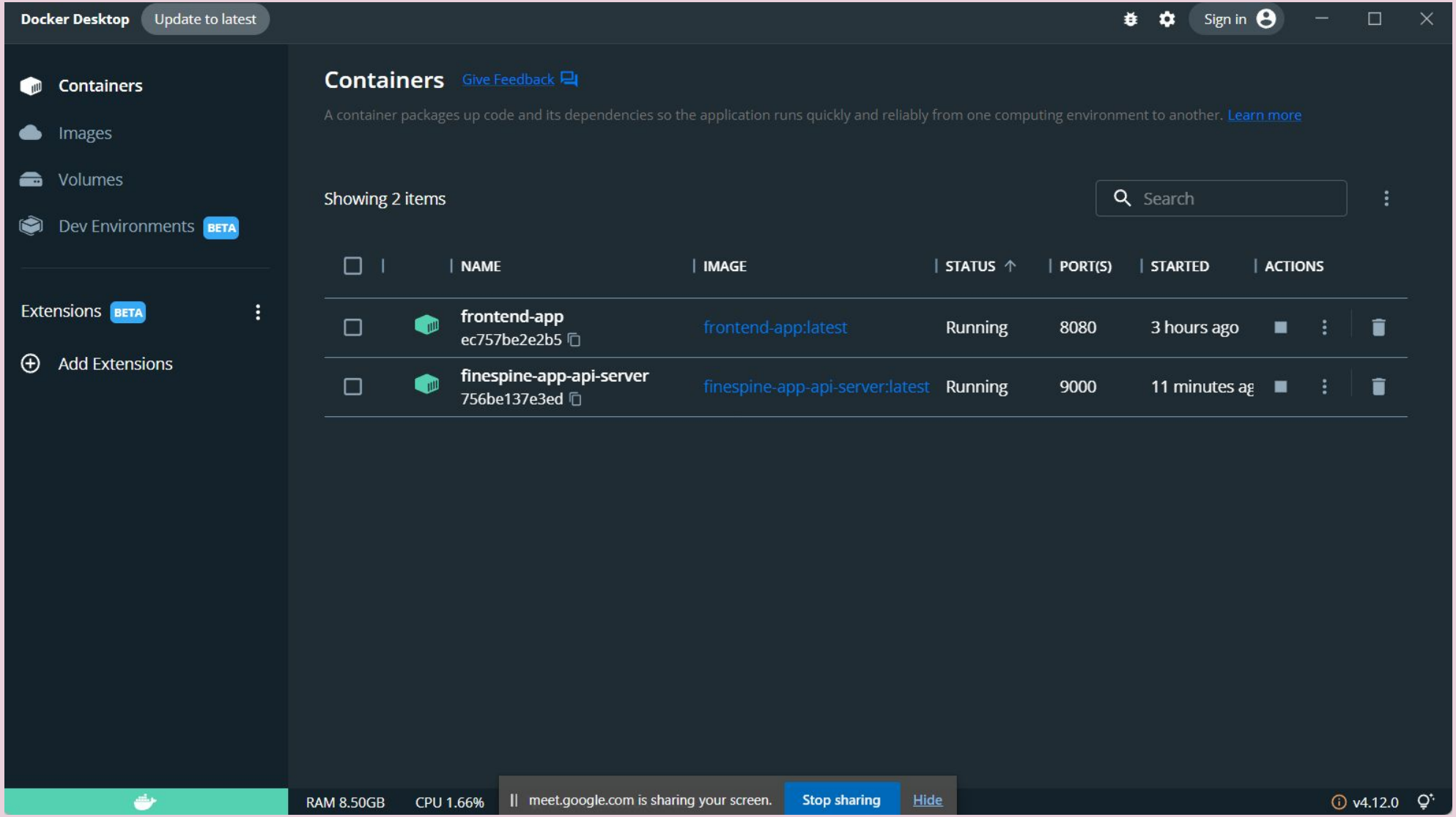



GCS Bucket
Data & Models
Terminal
Browser
REST / JSON
HTML
Download Images Container
API Service Container
Frontend App Container
Mount
Source Code
Mount
Source Code
Mount
Source Code
Mount
Mount
Persistent Folder
Secrets

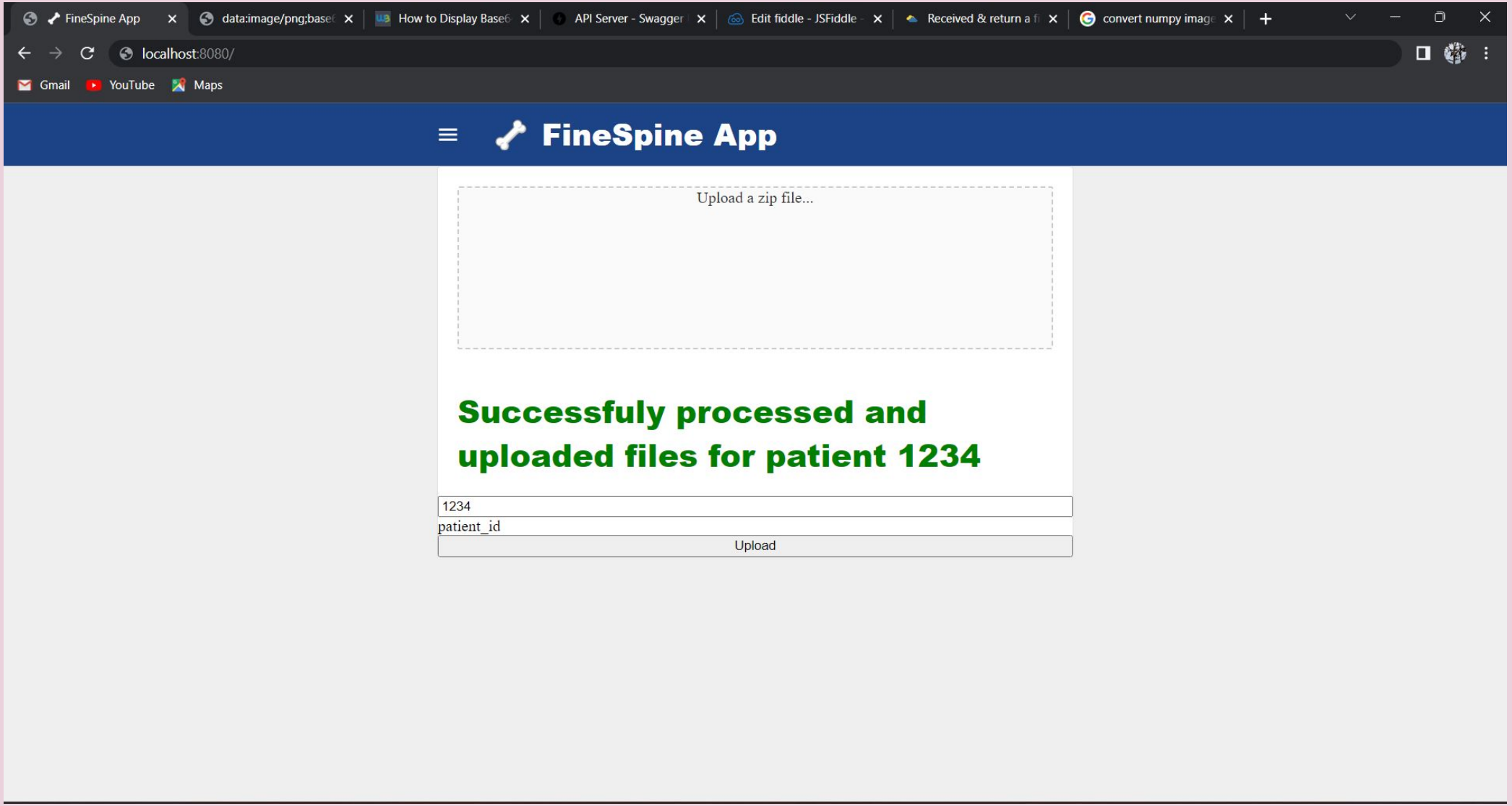
Wireframe Demo (Figma)

- We have created a simple wireframe for our FineSpine app using Figma.
- Go to the link
- Click on the clickable tabs (user/admin login) to navigate through the pages
- Press R to go back to the main page
- Click here

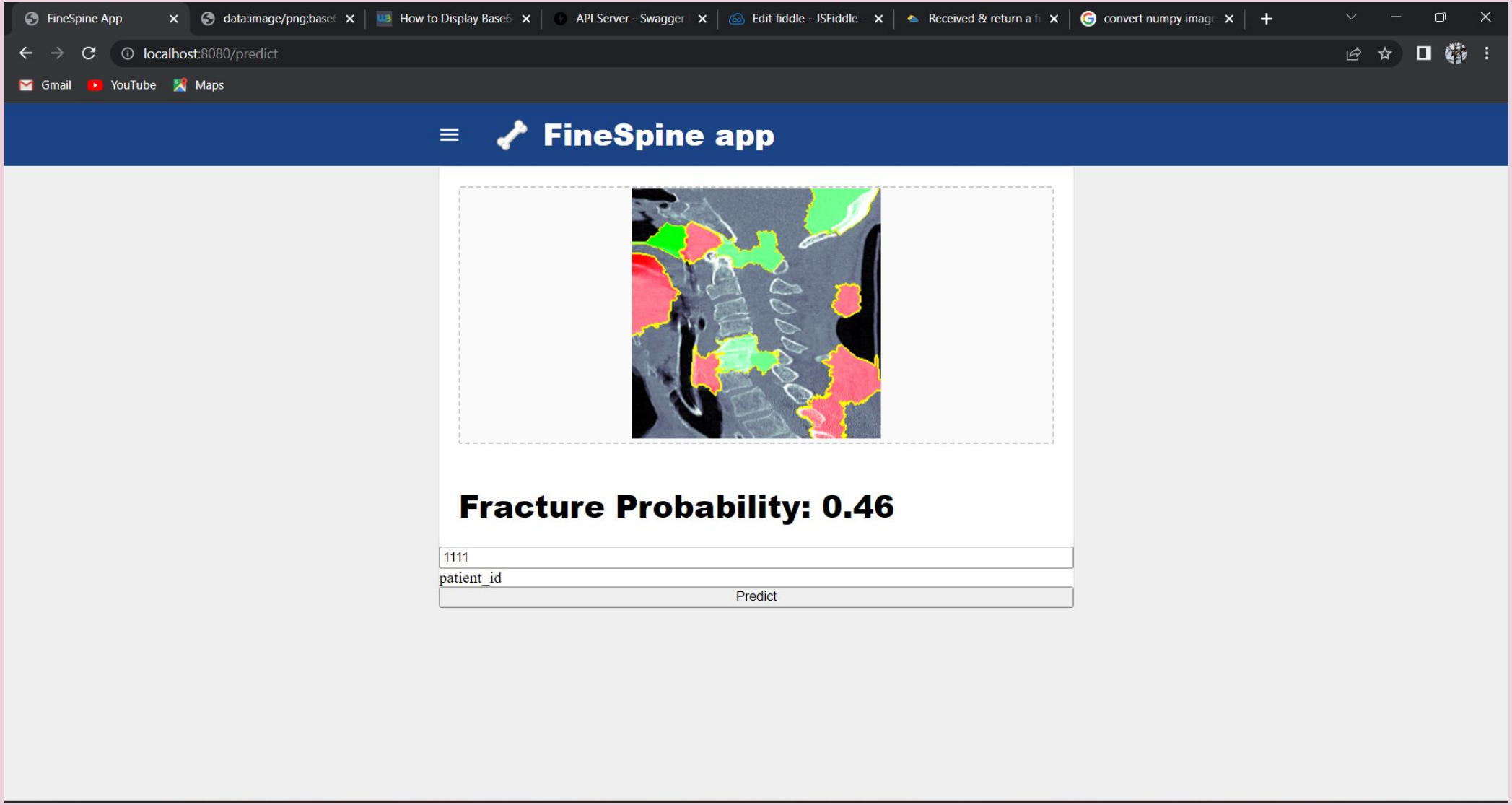
Running Containers (localhoast)



Upload Page (Bootstrap)

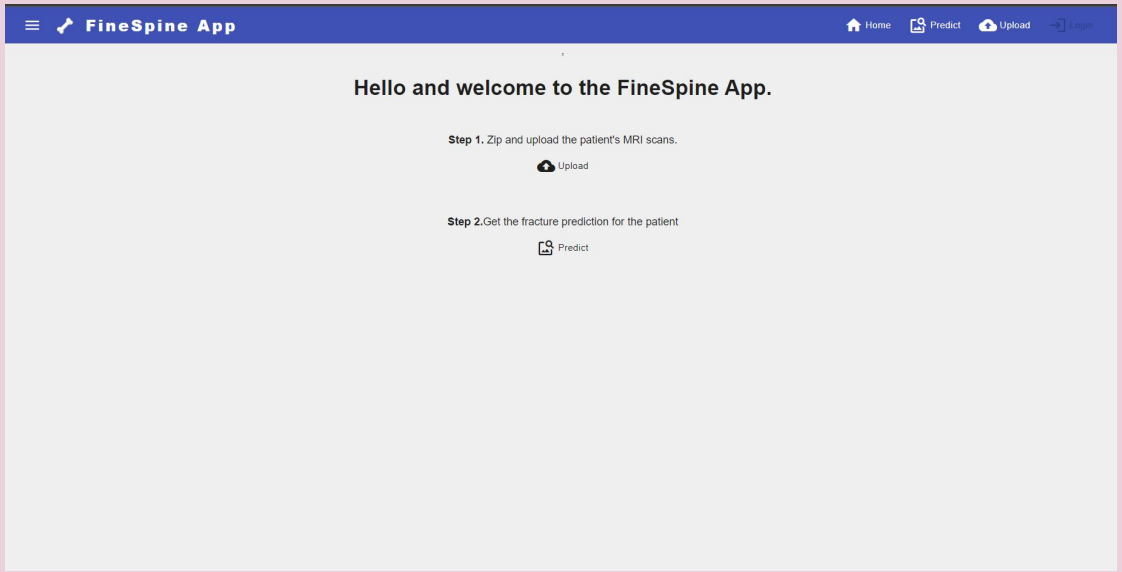


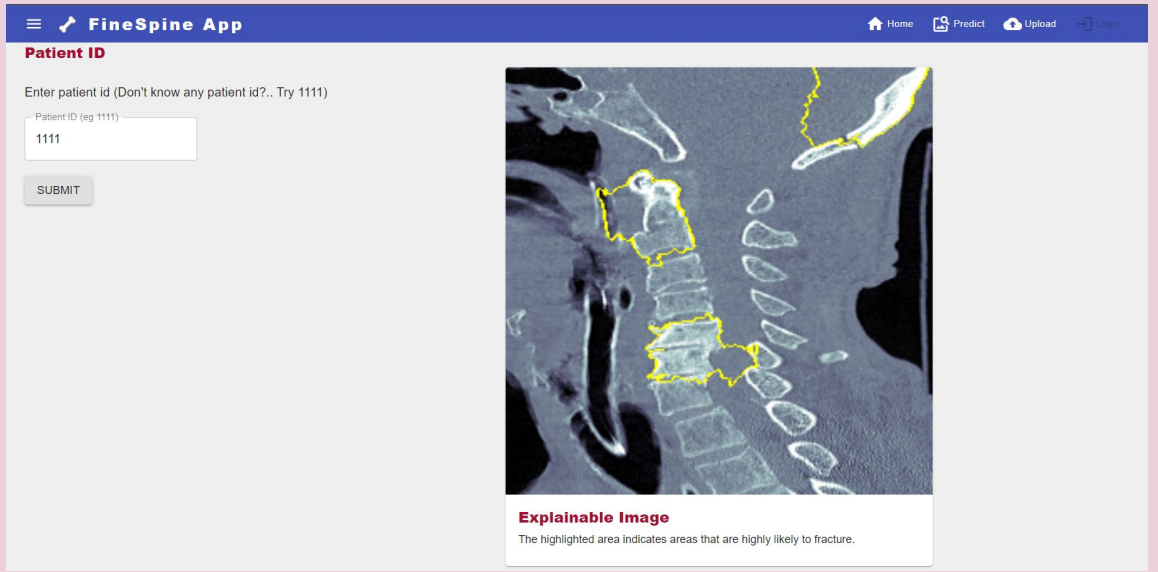
Predict Page (Bootstrap)



Production App

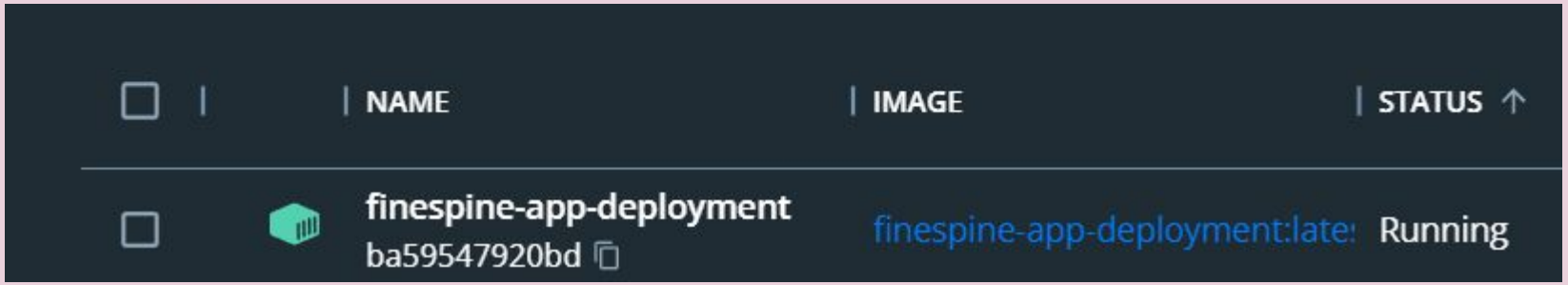
- We used React.js for the production.
- Component based framework
- High Performance





Deployment

- Create a standard docker container for deployment

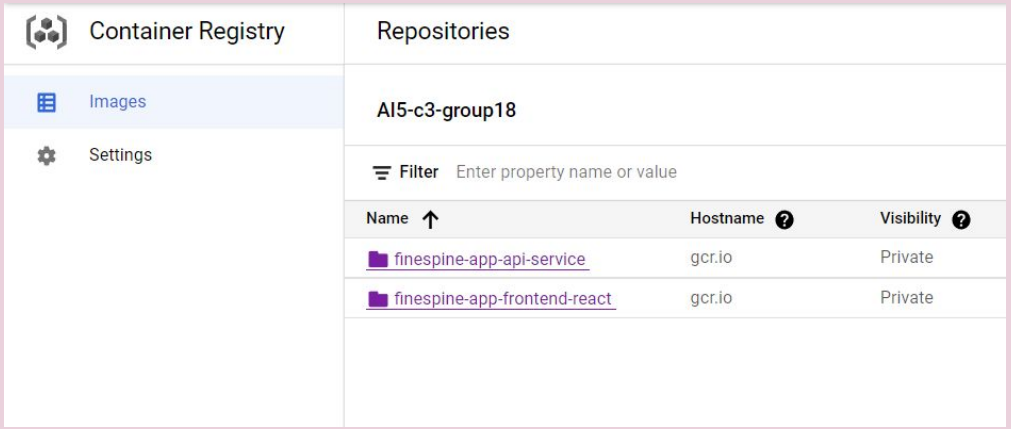


- Ansible to automate creating and running GCR, GKE

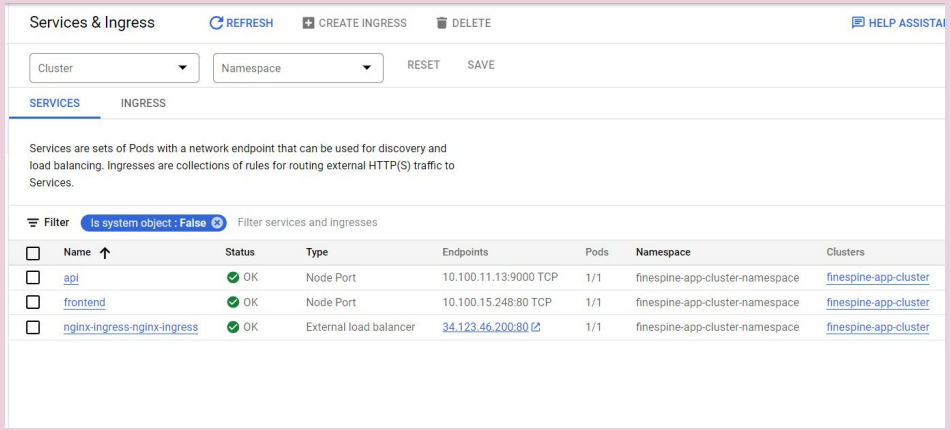
```yaml
- name: "Build docker images and push them to GCR"
  hosts: localhost
  gather_facts: false
  tasks:
  - name: "Get timestamp for docker tag"
    shell: "(date +%Y%m%d%H%M%S)"
    register: tag
  - name: Print tag
    debug:
      var: tag
```

Deployment (GCR and GKE)

- GCR for storing and versioning of the docker containers.



- GKE cluster for Auto Scaling and Self Service.



THANK YOU