HospitAPP

Contributors: Tejaswini Yeleswarapu, Arushi Agarwal, Jyoti Gambhir, Apurvi Mansinghka, Himanshu Kumar, Muhammed Yaseen Harris

Abstract: Coronaviruses are a large family of viruses which may cause illness in animals or humans. In humans, several coronaviruses are known to cause respiratory infections ranging from the common cold to more severe diseases. The most recently discovered coronavirus is due to COVID-19. The number of new cases are increasing day by day around the world and has been declared a pandemic by the World Health Organisation (WHO). In light of this global health emergency, countries all over the world are facing unfamiliar situations and have little estimation of the no. of crucial medical resources (like hospital beds, medical equipment, etc) needed over the course of the next few weeks. In order to efficiently enforce measures and policies to control the spread of the virus, governments need to be dynamically updated about the severity of the said spread. Our aim is to build a web app for the healthcare institutions and the government to keep track of the situation within a specific city. The healthcare institutions will directly log cases (new/active/recovered cases and deaths), and our machine learning model will predict the cases for the coming week. Leveraging this information, they can keep stock of resources (Beds, PPEs and ventilators) beforehand. Secondly, they can distribute said resources on priority basis, and productively divide them among the institutions, with high exposure hospitals getting a fair share. The government will continuously monitor the information/projections and ensure the availability of resources to the healthcare institutions. They can also ensure the preparedness of the other hospitals (with maybe low exposure, and in less risk zones) in case of a high surge prediction.

Objective

To build a web app to swiftly identify high risk zones based on our weekly predictions and thereby efficiently allocate medical resources (Beds, PPEs and ventilators) to the hospitals/healthcare institutions in those zones accordingly.

Goals

- To help healthcare institutions directly log cases and keep track of weekly projections to avoid congestion.
- To aid the government in keeping track of high and medium risk zones on a weekly basis:-
    - Allocate essential medical resources in advance to handle future rise in cases.
    - Redirect medical resources to handle present cases.
    - Promptly take measures in projected high/medium risk zones.
- Directly notify (pop up alert) the government when resources are running low, current high zones and projected high risk zones.
- Zone wise availability of information on cases, down to individual hospitals/healthcare institutions.
- Zone wise availability of information on the capacity of hospitals, down to the number of medical resources (Beds, PPEs and ventilators) in each hospital.

Target Users

Healthcare Institutions, Government

Scope

This web app is currently limited to hospitals in Hyderabad and the Telangana government.
It can easily be scaled to incorporate other cities as well.

Assumptions

- People will go to a nearby hospital (not necessarily the nearest one) , i.e, a hospital in the same zone as where they live. The zones will be marked as high, medium or low risk depending on the number of active cases, current rise in cases and/or projected rise/fall in cases reported by the hospitals in those zones.

Functionality

- Build an app for healthcare institutions to directly log on a daily basis
    - New cases, Active cases, recovered cases and deaths.
    - Present capacity, i.e, no of hospital beds, ventilators and PPEs (personal protective equipment).

- Use this data to make weekly projections of rise/fall in cases and mark zones within the city as high, medium or low risk based on this data for the government to:
    - Promptly seal off a high risk zone and impose precautionary measures for inhabitants of that zone.
    - Prepare in advance by either
        - redirecting essential medical resources from hospitals in low risk zones to the hospitals in the most affected zones, or
        - order additional equipment and allocate resources depending on the capacity of each hospital to avoid shortage of resources and equipment, and congestion.

Method

- Use data from hospitals to make a predictive model for weekly projections of cases.
- Weekly projections of new cases, deaths per zone.
- The city is divided into 30 zones based on the GHMC's division of zones.
- Zones are classified as high/medium/low risk based on no. of projected rise or fall in cases and deaths.

Technologies used:

A. Django Framework

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. It takes care of much of the hassle of Web development and follows the model-template-view architectural pattern.

B. MongoDB

MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schema. It supports field, range query, and regular-expression searches. Queries can return specific fields of documents.

C. Docker (Used for simulating Distributed App)

Docker is a set of platform as a service products that uses OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files. They can communicate with each other through well-defined channels.

Reasons for using Docker in the web app:

- ❏ Simulating a distributed app

- ❑ Setup once done in an image (image is a running container in Docker), can be shared to others, which saves them from having to do the entire setup on their own systems. This greatly helped us while working as a remote team.

Features

- Home page:-
  - **Description:** First page that the user will see on opening the app.
  - **Purpose:** For the user to login/sign-up to the app.

- Login:-
  - **Description:** Unique username and password
  - **Purpose:** To ensure that only authentic users can access/input the valid information.

- Page after login:-
  - *Government*
    - **Description:** Government users after login will be redirected to the statistics and heatmaps page.
  - *Hospitals*
    - **Description:** Hospital users after login will be redirected to the hospital information entry page.

- Hospital Information:-
  - **Description:** Capture basic information of hospitals, bed capacity and location (zone).
  - **Purpose:** To segregate hospitals into various zones.

- Logging in Data:-
  - *Cases*
    - **Description:** Record new, active, recovered cases and deaths daily.
    - **Purpose:** To update the government on a daily basis and to use for weekly projections of cases.
  - *Medical resources and equipment*
    - **Description:** Record total number of beds/ventilators/PPEs along with how many currently in use and how many available on a daily basis.

        - **Purpose:** To immediately alert the government to allocate resources (according to hospital capacity) based on our weekly projections (zonal level) and direct resources on a daily basis if a hospital is running out of essential equipment despite having the capacity to treat more patients.

- Statistics:-
    - **Description:** No. of active/new/recovered cases and deaths in each zone on a daily basis as well as weekly predictions.
    - **Purpose:** To be aware of the situation on a daily basis and prepare in advance based on the weekly predictions.

- Heat Maps: -
    - **Description:** Zone wise heatmaps to show which zones currently have the most (and least) cases and classify them as high (red), medium (yellow) or low risk (green) along with projections in a week's time.
    - **Purpose:** Giving the government information to take necessary precautionary measures.

- Pop up Notification:-
    - *Cases*
        - **Description:** Pop up alert on both the government and hospital pages when a particular zone/zones are projected to be high risk or are currently high risk.
        - **Purpose:** Instant alerts for the government and hospitals in that zone to take immediate action.

    - *Medical equipment*
        - **Description:** Pop up alert on the government and hospital pages when a hospital (depending on the capacity) within a zone or all the hospitals in a zone need more medical equipment for current and projected cases.
        - **Purpose:** Instant alerts for the government and the hospitals in that zone (or a nearby zone if all the hospitals in that zone have reached their limit or if that zone has very few hospitals) in advance to share the load if some hospitals have already reached their limit, and direct essential medical care accordingly.

    - *General Public (if time permits)*

        - Send sms to the inhabitants of a high risk/projected high risk zone from the government's end.
        - Pop up alerts and other features

Implementation

A. App Interface

Since the app targets both the healthcare institutions as well as the government, the interface for the two will have slight variations:-

- Healthcare Institutions/Hospitals
    - Homepage and login will be common
    - Users will be redirected to the hospital information entry page.
    - Once done, users can then log patient data, equipment details, see weekly projections and put in a request for resources from the government
    -
- The Government
    - Homepage and login will be common
    - After login, users can see zonal data, weekly projections, individual hospital data, zonal data and view/accept/reject the request for resources from the healthcare institutions
    - The government cannot update or log data.

B. Machine Learning Model

- ***Working of the Model:***
    - The model takes in 22 days worth of data and predicts the next 7 days. For the sake of the demo, we're assuming that we already have the necessary data.
    - The model predicts
        - Number of new cases in the next 7 days
        - Number of deaths in the next 7 days
        - Classification of zones as high, medium or low risk based on point 1.
    - We built a model from scratch and used a new dataset.

- ***Dataset:***
    - The dataset is a combinationation of stationary features and sequential features.
    - Since the aim is to predict the spread of COVID-19, for better accuracy we also took into account demographic factors (census) along with the data from the hospitals.

- The stationary data was procured from the US census website. We employed feature engineering to carefully pick out the features that could potentially affect the spread. It consisted of race, total population, population divided by age and sex and income with a total of 76 columns. (This can be replaced by indian data)
- The sequential data was taken from Kaggle and consisted of the date, zone location, current new cases and current deaths with a total of 4 columns.
- We had data for 670 counties:-
  - Testing - 570
  - Validation - 30
  - Testing - 30

- ***Preprocessing:***
  - The Kaggle dataset (sequential) had cumulative cases, instead of just new cases. In order to train the model, we had to subtract each row from its preceding row. (for ex, if the total no. of cases on day 1 is 13 and on day 2 is 16, the number of new cases on day 2 is 16-13 = 3)
  - We replaced the date format with an integer. (ex, 14/4/20 = day 1 and so on)
  - We then combined the sequential features with the stationary features.
    - No of stationary features = 76
    - No of sequential features = 2 per model (date and new cases for cases prediction, date and deaths for deaths prediction)
    - Total number of features per zone = 22x78 = 1716
  - Finally, we scaled the dataset to 0 mean and unit variance, a standard ML procedure.

- ***Model:***
  - We used an LSTM because it handles sequential data well.
  - The model is essentially a sequence:-

  - We employed batch processing to parse data into the model
    - We divided the 570 samples in batches of size 64
  - *Input Layer*: Each of the 64 samples had 22 timesteps and each timestep had 78 features.
  - *LSTM*: The output of the previous layer was passed to an LSTM of hidden size 120.

- *Output Layer:* The output from the LSTM is passed into the final linear layer that produces the final output, i.e, no. of new cases and deaths.

- ***Accuracy:***
  - Mean absolute error of cases: 12.5
  - Mean absolute error of deaths: 0.8
  - The error rate for deaths is low because the model is more accurate when the numbers are low.
  -

C. Database (MongoDB)

- **Input Data:**
  - Hospital data entry
    - **Registration :**
      - Name
      - location
      - total bed capacity
      - Each row will correspond to a different hospital.
      - Every time a new hospital registers, this database gets updated.

    - **Daily update logging (excel sheet no. 2):**
      - Name
      - location
      - Date
      - Medical resources info
        - current no. of beds,
        - current used beds,
        - current number of ventilators,
        - currently used ventilators,
        - PPE stock (1 PPE unit = 1 mask+2 gloves),
      - Cases Info
        - currently active cases,
        - New cases
        - number of people discharged (recovered),
        - number of deaths

    - **Zonal (daily logging):**
      - Zone Name
      - Date
      - Medical Resource Info

        - current no. of beds,
        - current used beds,
        - current number of ventilators,
        - currently used ventilators,
        - PPE stock (1 PPE unit = 1 mask+2 gloves),
    - Cases Info
        - currently active cases,
        - Number of new admits,
        - number of people discharged (recovered),
        - number of deaths.
    - Each row will correspond to the total no. of cases/medical equipment by hospitals in that zone.

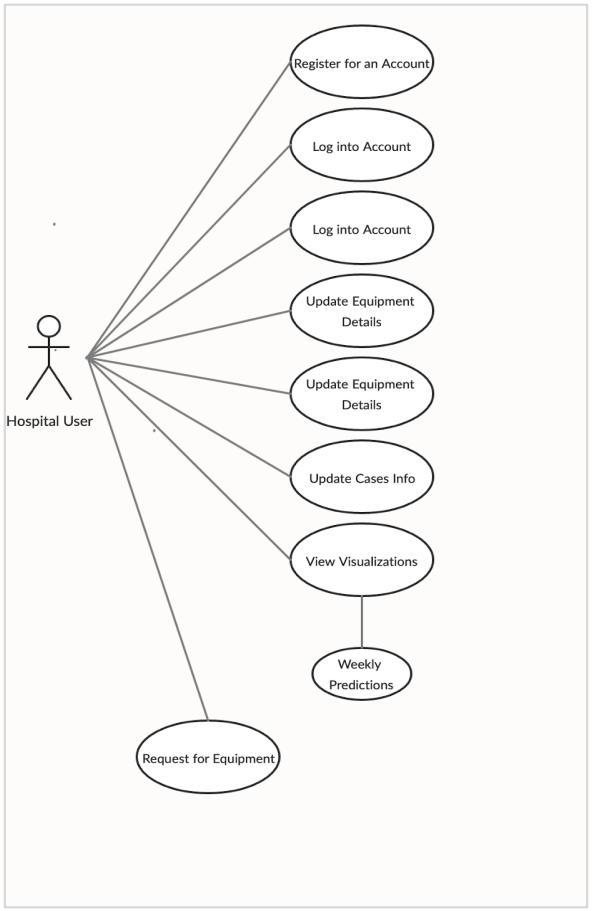
- **Output Details:**
    - Total: 17 columns
    - Column 1: location (pincode)
    - Column 17: location (name of the zone)
    - Column 2-8: predictions of new cases for the next 7 days
    - Column 9-15: predictions of deaths for the next 7 days
    - Column 16: Classification of zones as high, medium or low risk
    - Rows will correspond to number of zones
    - Will be updated once a day at midnight.

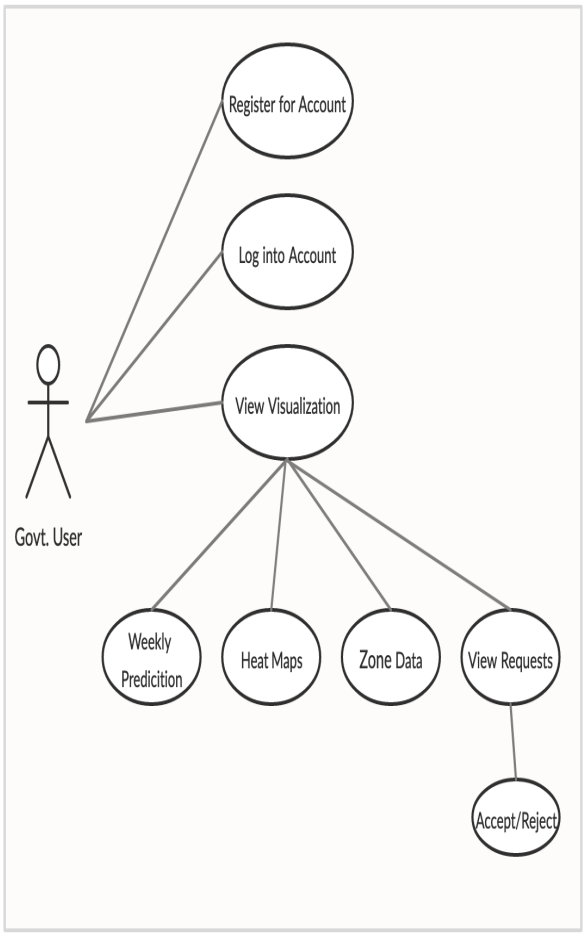
Feature Comparison

The following table represents a comparison of the final features against our initial features:-

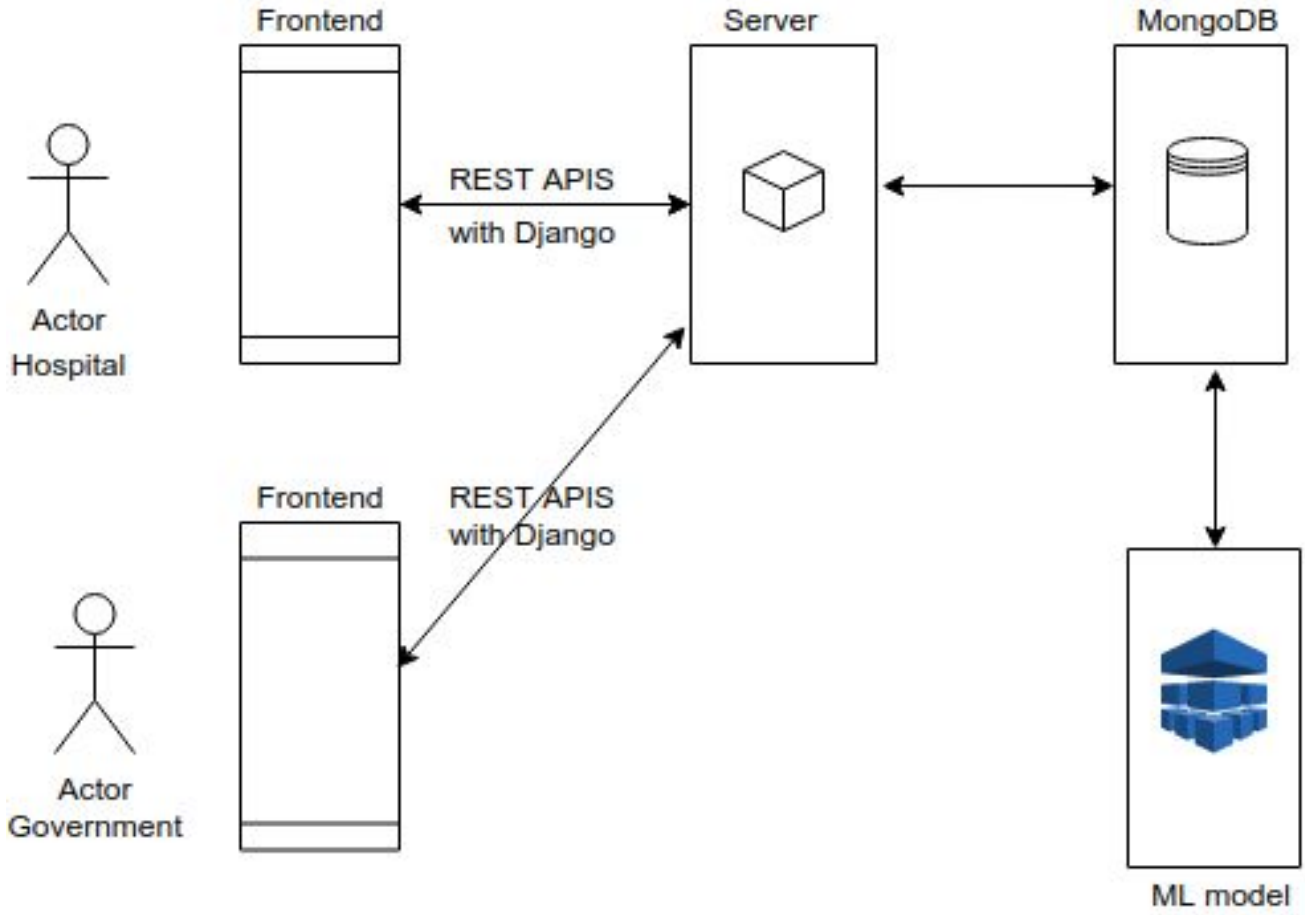
Initial Features	Final Features
Register for an Account	Register for an Account
Login	Login
Daily log of cases and medical equipment (healthcare facility's end)	Daily log of cases and medical equipment (healthcare facility's end)
Statistics of daily Data	Statistics of daily Data
Statistics of *weekly active* cases	Statistics of *weekly new* cases
Heatmap for all *30 sub-zones*	Heatmap for *5 main zones*
Pop-up notification alert for high risk zones	*Request form* for medical equipment

Use Case Diagram

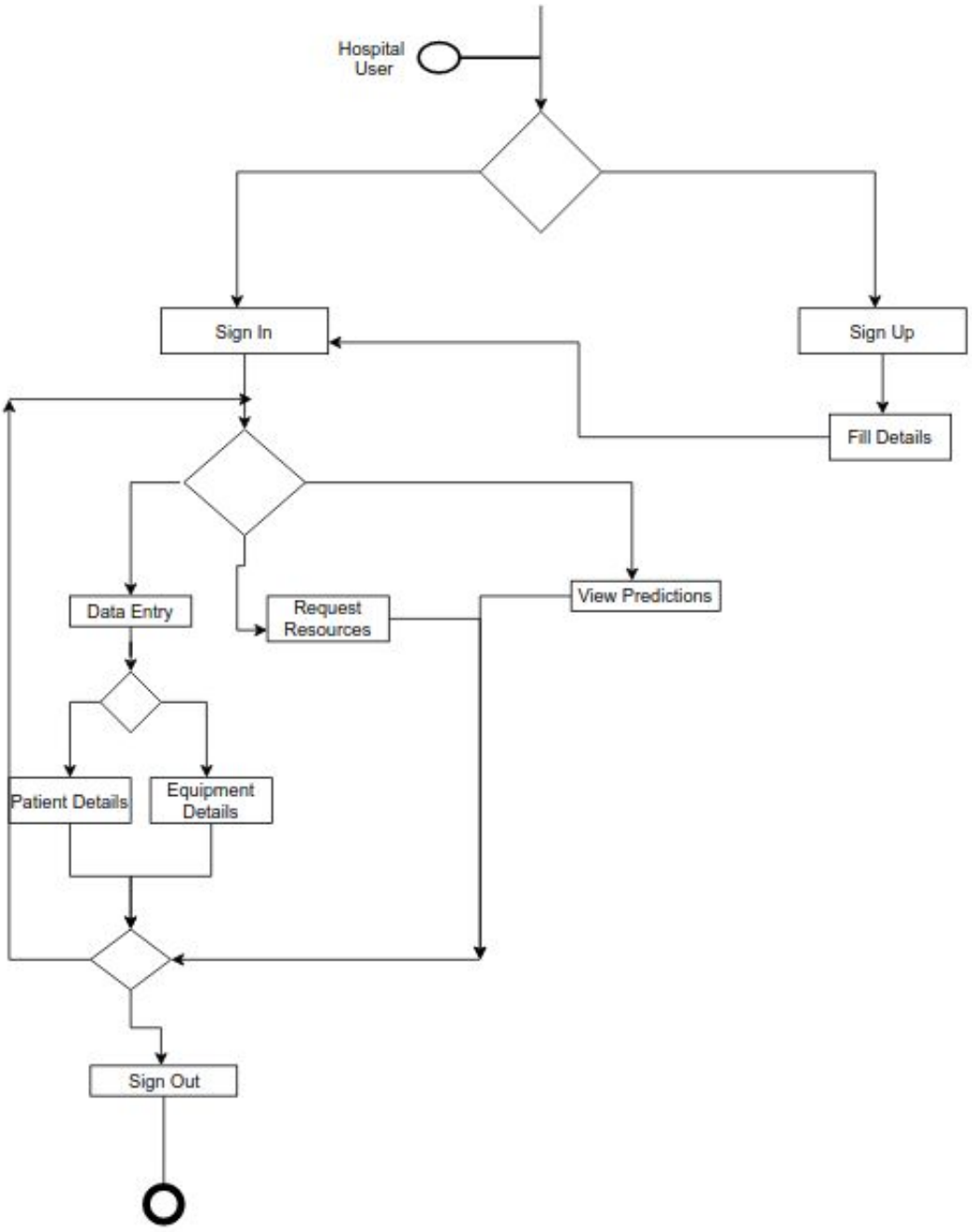


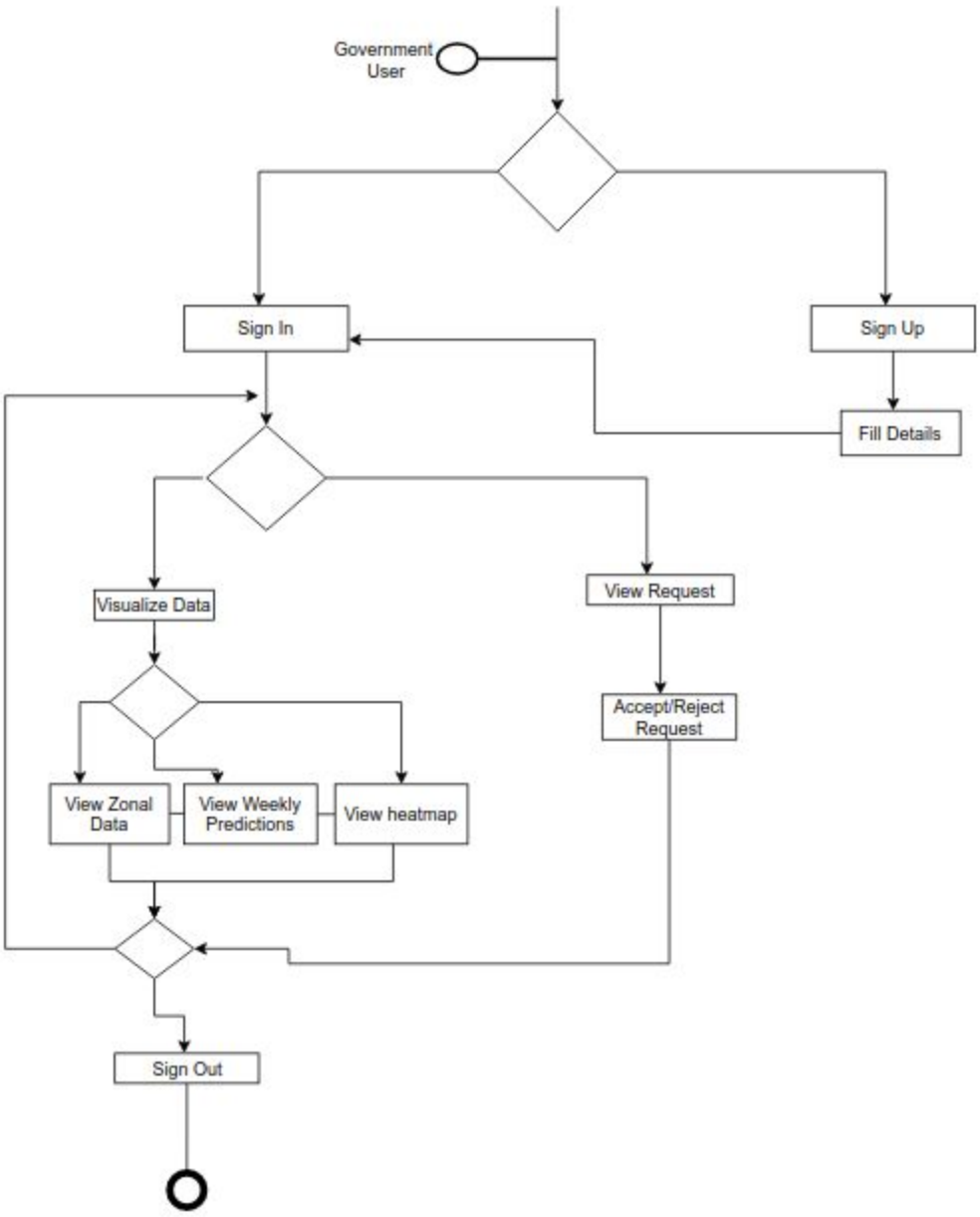


High Level Diagram



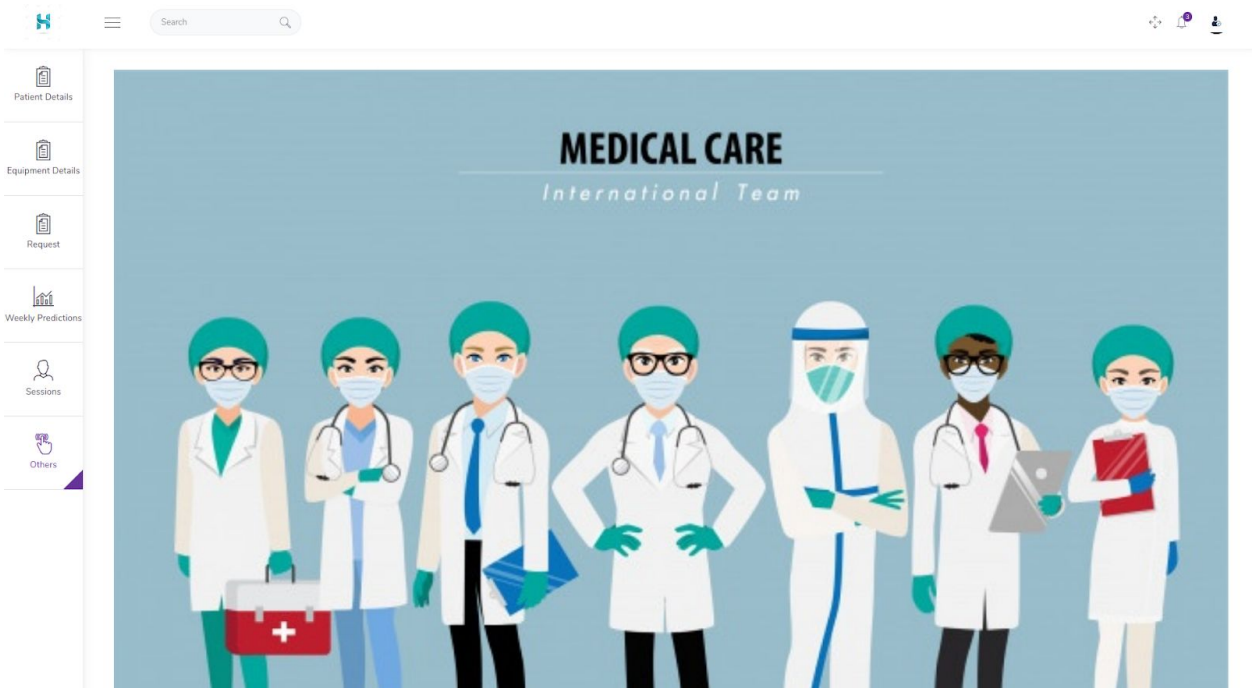
Activity Diagram




Government User
Sign In
Sign Up
Fill Details
Visualize Data
View Request
Accept/Reject Request
View Zonal Data
View Weekly Predictions
View heatmap
Sign Out

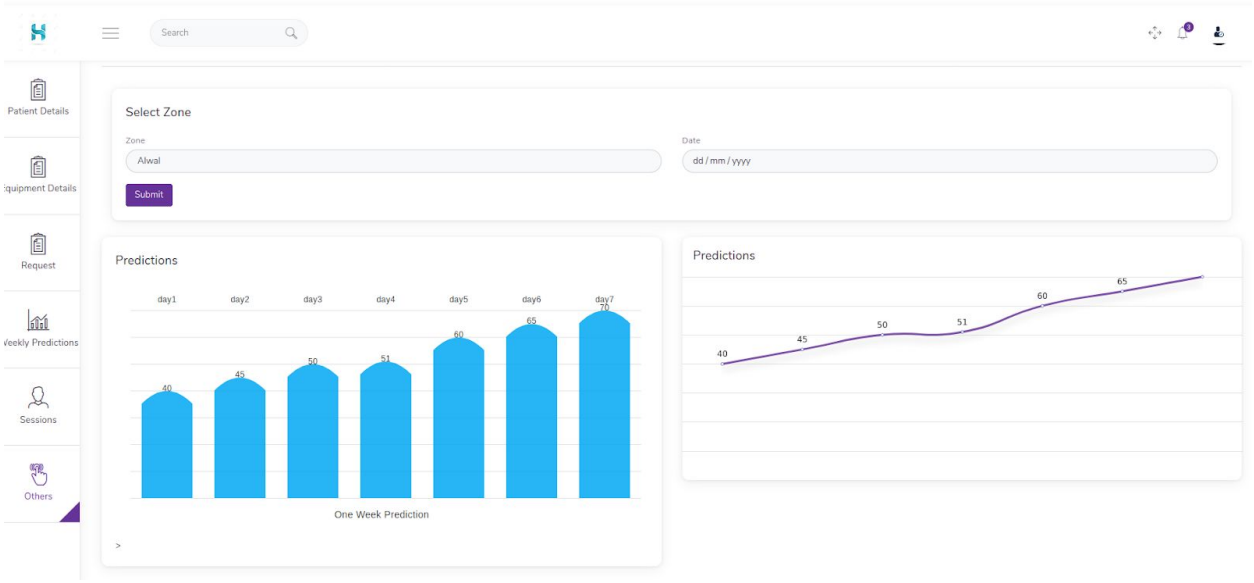
Snippets from HopsitAPP:

A. Hospital Home Page

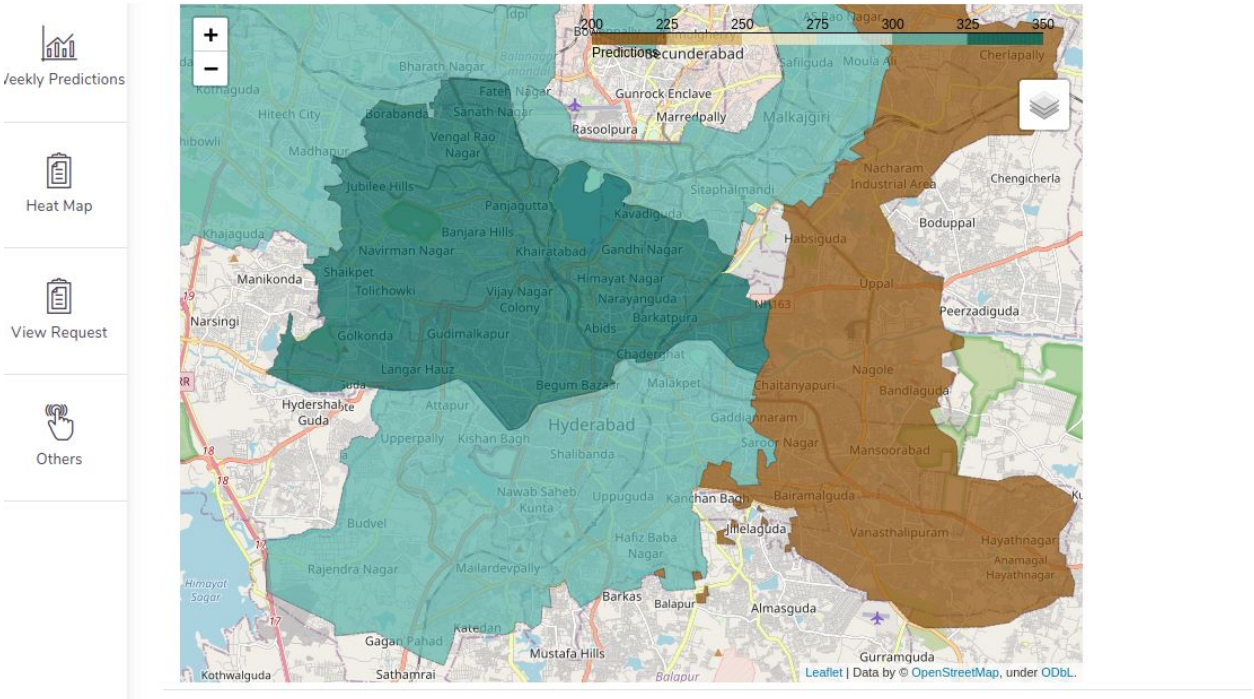


B. Government HomePage

C. Weekly Prediction



D. Heatmaps



E. Predicted Risk Zones

Prediction For the Next Week

Zone	Predicted Risk
Alwal	low
Amberpet	medium
Begumpet	low
Chandanagar	medium
Chandrayangutta	low
Charminar	low
Falakunuma	medium
Gajula Ramaram	low
Goshamahal	high
Hayathnagar	low
Jubilee Hills	medium
Kapra	low
Karwan	low
Khairatabad	low
Kukatpally	low

F. Zonal Data Visualization

Future Work:

- Predicting farther into the future. (ex: 1 month, 3 month)
- Pop up notification on the healthcare user end when their request has been approved/disapproved by the government
- Sending out SMS/Alerts to the general public in a Red Zone
- Scaling the app to include more cities/states/countries etc.