High Level Design (HLD)

Deep EHR: Chronic Disease Prediction Using Medical Notes

Revision Number: 1.0
Last date of revision: 15/06/2021

Asif Khan

Document Version Control

Date Issued	Version	Description	Author

Contents

Abstract

Healthcare organizations have aimed to improve the quality of patient health care and required access to the patient medical information in no time that has been made possible through the introduction of EHR (Electronic Health Record).

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3 Definitions

Term	*Description*
EHR	Electronic Health Record
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment
AWS	Amazon Web Services

2 General Description

2.1 Product Perspective

The Deep EHR system is a web application which will detect the diseases at earliest for better disease management, improved interventions, and more efficient health-care resource allocation using previous EHR records available. SQL is used to retrieve, insert, delete, and update the database. Here the system store each and every data given by user or received in request to the MySQL/MongoDB database.

2.2 Tools used

Python programming language and frameworks such as Numpy, Pandas, Scikit-learn, TensorFlow, Keras are used to build the whole model.

- PyCharm is used as IDE.
- For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
- AWS is used for deployment of the model.
- Tableau/Power BI is used for dashboard creation.
- MySQL/MongoDB is used to retrieve, insert, delete, and update the database.
- Front end development is done using HTML/CSS
- Python Django is used for backend development.
- GitHub is used as version control system.

2.3 Constraints

The Deep HER system must be user friendly, as automated as possible and users should not be required to know any of the workings.

2.4 Assumptions

The main objective of the project is to predict the onset of disease for new cases based on the information in the EHR by using Machine Learning and Deep Learning techniques. It is also assumed that all aspects of this project have the ability to work together in the way the designer is expecting.

3 Design Details

3.1 Functional Architecture

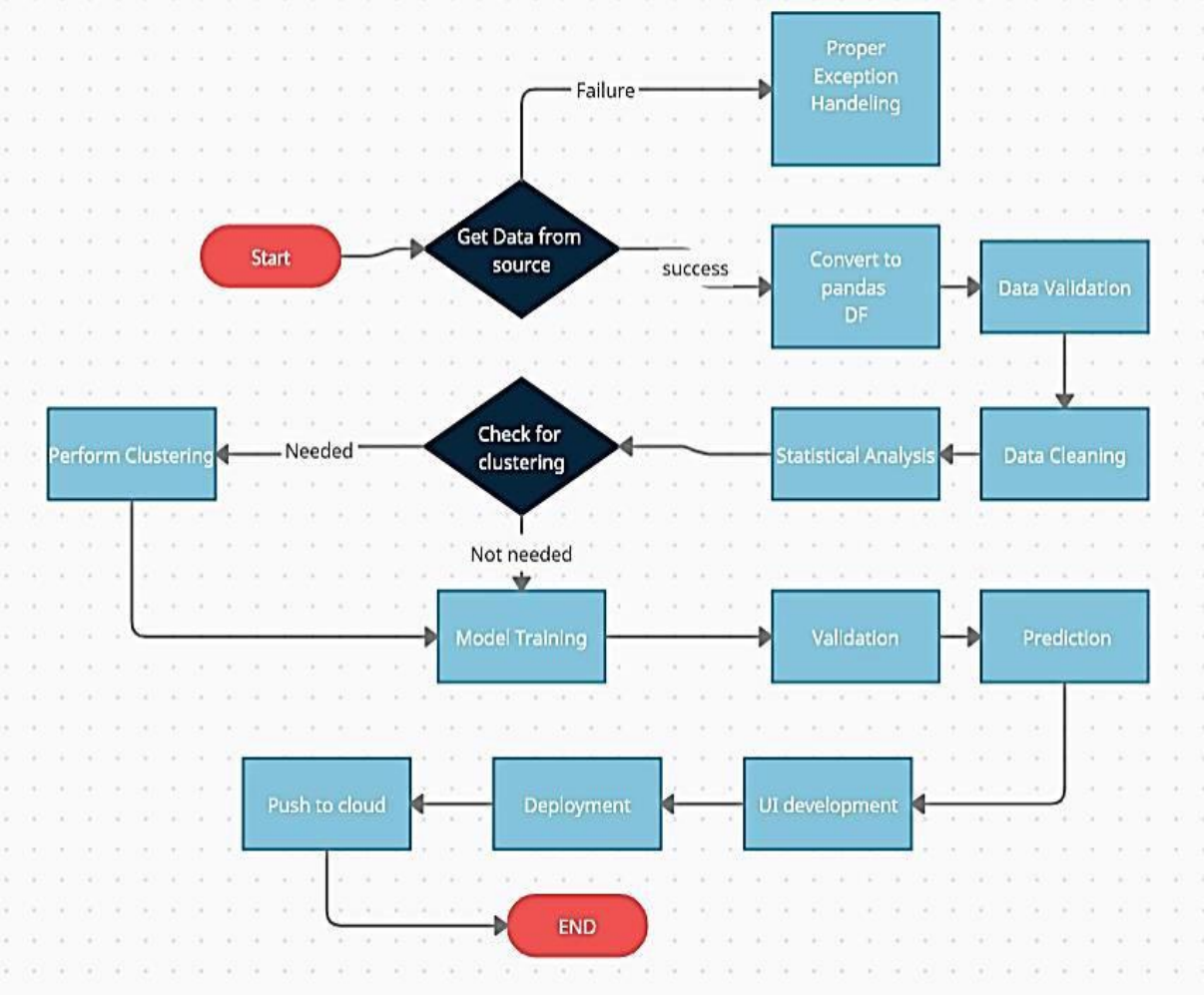


Figure 1: Functional Architecture of Deep EHR

3.2 Database Design

System needs to store every request into database and you need to store in such a way that if you want to retrain a model it should be easy to retrain model with new data as well.

Initial Step-By-Step Description:

- The User chooses the disease.
- The User gives required information.
- The system stores each and every data given by user or received in request to the database.

3.3 Web Application Architecture

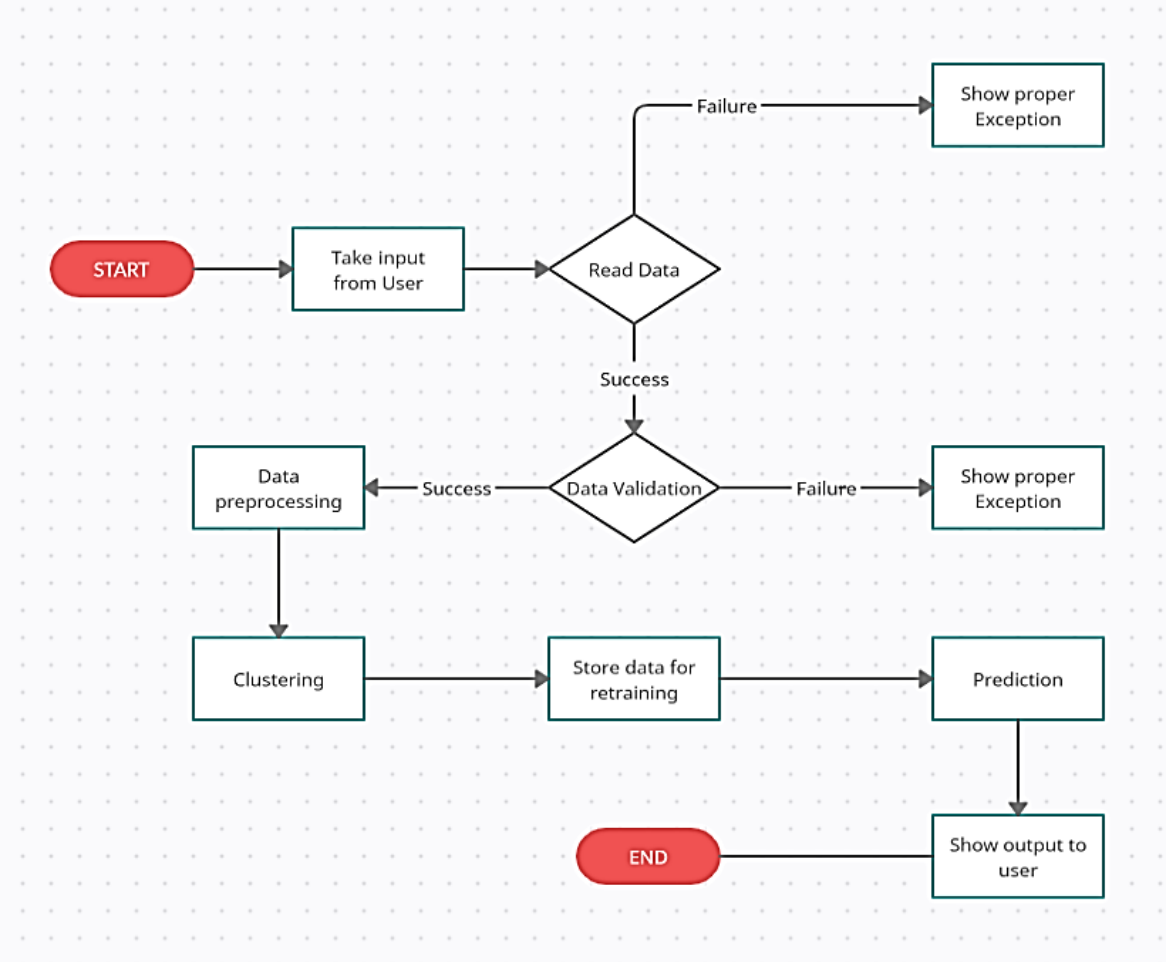


Figure 2: Web application architecture

The user interface is a very simple plain layout with little graphics. It will display information very clearly for the user and will primarily output information to the user through HTML pages. Also, all the details for the user input will be provided.

3.4 Event log

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

- The System identifies at what step logging required
- The System should be able to log each and every system flow.
- Developer can choose logging method. You can choose database logging/ File logging as well.
- System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.5 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong and the error will be defined as anything that falls outside the normal and intended usage.

3.6 Help

The 'Help' option is provided in web application for guiding users regarding maximum range of valid inputs required for predicting a particular disease.

3.7 Performance

Deep EHR system is used for predicting the disease onset and should be as accurate as possible so that it will not mislead the user. Also, model retraining is very important to improve the performance.

3.8 Security

Since the Deep EHR system consists of patient's demographic data, the information is secure.

3.9 Reusability

The code written and the components used should have the ability to be reused with no problems.

3.10 Application compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

3.11 Resource utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

3.12 Deployment

4 Dashboards

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the disease.

As and when the system starts to capture the historical/periodic data for a user, the dashboards will be included to display charts over time with progress on various indicators or factors.

4.1 KPIs (Key Performance Indicators)

- Key indicators displaying a summary of the person's health and physique as compared to a normal individual with similar basic traits.
- Time and workload reduction using the EHR model.
- Comparison of accuracy of model prediction and doctor's prediction.
- Number of times a patient visits the hospital.
- Time between symptom onset and detection of illness/visit to hospital.
- Immunity of patient (based on previous illnesses).
- Vaccines the patient has taken.
- Length of stays in hospital.

5 References

Deep EHR: Chronic Disease Prediction using Medical Notes' by Jingshu Liu, Zachariah Zhang and Narges Razavian

Low Level Design (LLD)

Deep EHR: Chronic Disease Prediction Using Medical Notes

Revision Number: 1.2

Asif Khan

Document Version Control

Date Issued	Version	Description	Author

Contents

Abstract

Healthcare organizations have aimed to improve the quality of patient health care and required access to the patient medical information in no time that has been made possible through the introduction of EHR (Electronic Health Record).

1 Introduction

1.1 Why this Low-Level Design Document?

The purpose of this document is to present a detailed description of the Deep EHR System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the higher management for its approval.

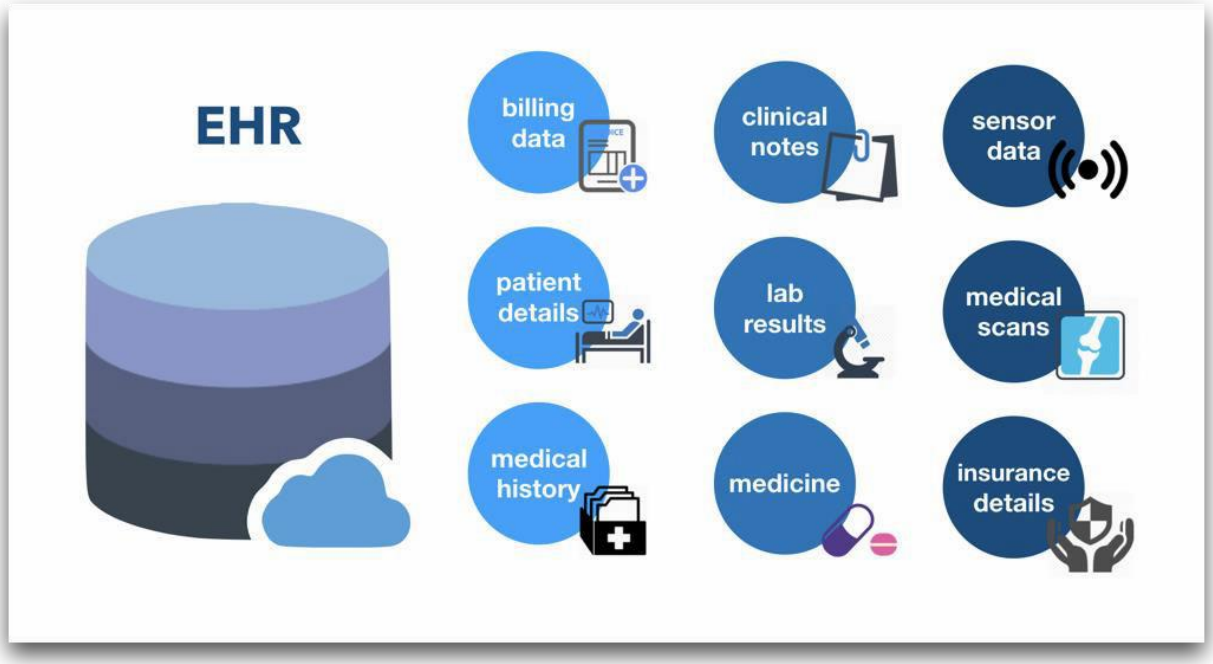
The main objective of the project is to predict if a person can get a chronic disease in his/her future based on the EHR, which stands for Electronic Health Record. EHR is nothing but a dataset of medical history of the patients.

EHRs are a vital part of health IT and can:

Contain the patient's past medical history, current diagnoses, current medications and treatment plans, immunization status, drug and environmental allergies, radiology images, and laboratory data.
Allow access to evidence-based tools that providers can use to make decisions about the patient's care
Automate and streamline provider workflow

An electronic health record (EHR) contains patients' health information such as:

- Patient demographics.
- Consultations, History and Physical, Operative Reports, Followup notes and Progress notes.
- Past Medical History.
- Past Surgical History.
- Social History.
- Family History.
- Physical Examination that includes General Appearance, Vital Signs that include blood pressure, heart rate, respiratory rate and temperature. Systems examinations include HEENT, pulmonary, cardiovascular, abdominal, neurological, dermatological exam and extremities exams.
- Current Diagnoses.
- Current Medications.
- Immunization Status
- Environmental and Drug Allergies.
- Radiology images
- Laboratory Data.



This project shall be delivered in two phases:

Phase 1: All the functionalities with PyPi packages.

Phase2: Integration of UI to all the functionalities.

1.2 Scope

This software system will be a Web application. This system will be designed to detect the diseases at the earliest for better disease management, improved interventions and more efficient health-care resource allocation using previous EHR records available, specifically early detection of any preventable diseases is important for better disease management. This system is designed to predict the diseases from the patient's information such as demographics, disease history, lab results, procedures and medications.

1.3 Constraints

We will only be selecting few of the chronic diseases.

1.4 Risks

Document specific risks that have been identified or that should be considered.

1.5 Out of Scope

Delineate specific activities, capabilities, and items that are out of scope for the project.

2 Technical specifications

2.1 Dataset

Disease	Finalized	Source
Diabetes	Yes	https://github.com/kthouz/Diabetes-PracticeFusion/tree/master/agg_data
Stroke	Yes	
Heart Disease	Yes	
Cancer	Yes	

2.1.1 Diabetes dataset overview

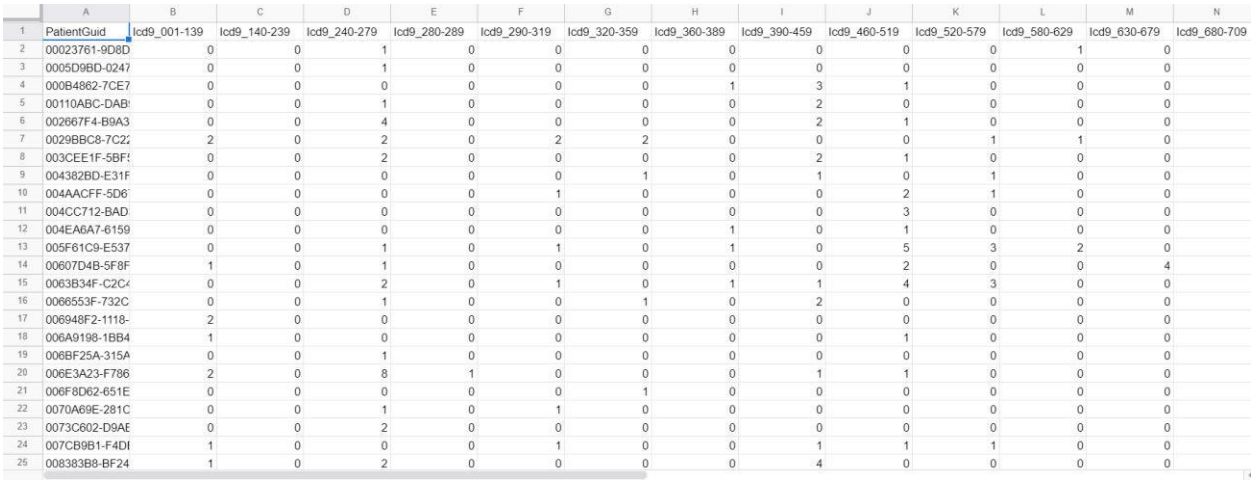
Consists of 4 different tables, Patient table consists of the patient's personal information and most importantly we have the historic data of a patient in the table diagnosis. Whereas the transcript table consists of patient demographic data. Physician_speciality table consists of patient behaviour.

There are a total of 9,948 patients in the training set and 4,979 patients in the test set.

- Patient table

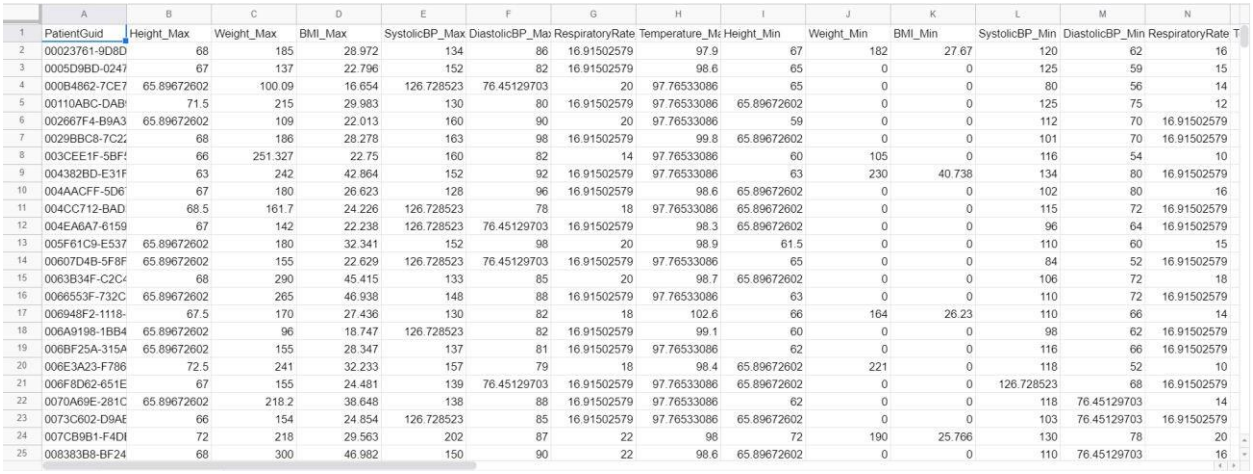
	PatientGuid	Age	Gender	State	DMIndicator
2	FB6EFC3D-1A2	83	1	41	0
3	C6746626-6783	27	0	43	0
4	E05C6E8F-779F	28	0	31	0
5	EAEBD216-F84	53	1	35	0
6	C7F10A80-4934	22	0	9	0
7	3BDB6A99-A40	65	0	9	0
8	5DA96225-0D61	43	1	35	0
9	D2516A79-B6E2	61	0	36	1
10	B06DEAB9-9F1	88	0	1	1

- Diagnosis table



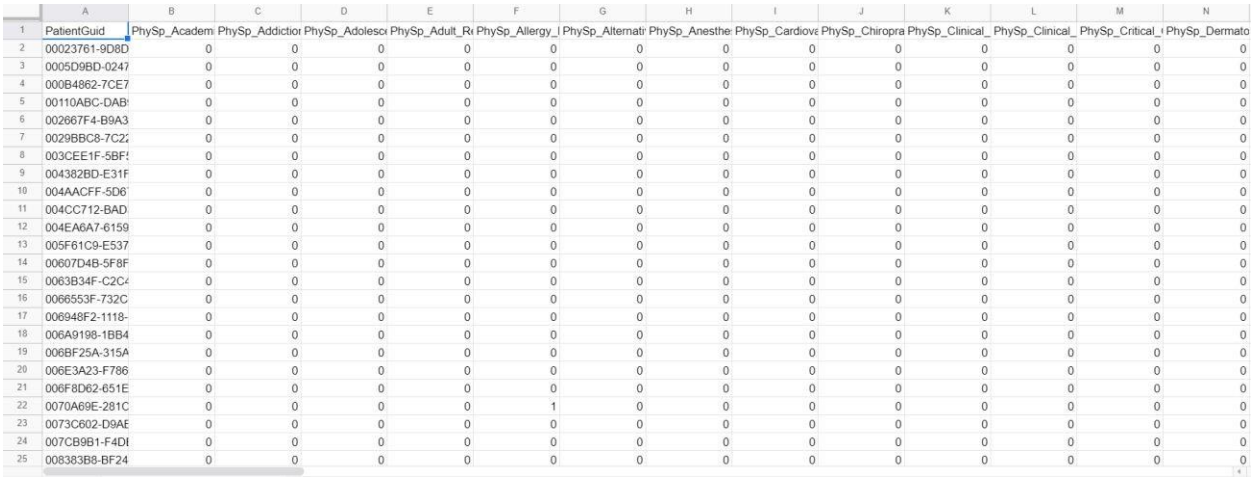
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	PatientGuid	lcd9_001-139	lcd9_140-239	lcd9_240-279	lcd9_280-289	lcd9_290-319	lcd9_320-359	lcd9_360-389	lcd9_390-459	lcd9_460-519	lcd9_520-579	lcd9_580-629	lcd9_630-679	lcd9_680-709
2	00023761-9D8D	0	0	1	0	0	0	0	0	0	0	1	0	
3	0005D9BD-0247	0	0	1	0	0	0	0	0	0	0	0	0	
4	000B4862-7CE7	0	0	0	0	0	0	1	3	1	0	0	0	
5	00110ABC-DAB	0	0	1	0	0	0	0	2	0	0	0	0	
6	002667F4-B9A3	0	0	4	0	0	0	0	2	1	0	0	0	
7	0029BBC8-7C22	2	0	2	0	2	2	0	0	0	1	1	0	
8	003CEE1F-5BF	0	0	2	0	0	0	0	2	1	0	0	0	
9	004382BD-E31F	0	0	0	0	0	1	0	1	0	1	0	0	
10	004AACFF-5D6	0	0	0	0	1	0	0	0	2	1	0	0	
11	004CC712-BAD	0	0	0	0	0	0	0	0	3	0	0	0	
12	004EA6A7-6159	0	0	0	0	0	0	1	0	1	0	0	0	
13	005F61C9-E537	0	0	1	0	1	0	1	0	5	3	2	0	
14	00607D4B-5F8F	1	0	1	0	0	0	0	0	2	0	0	4	
15	0063B34F-C2C4	0	0	2	0	1	0	1	1	4	3	0	0	
16	0066553F-732C	0	0	1	0	0	1	0	2	0	0	0	0	
17	006948F2-1118-	2	0	0	0	0	0	0	0	0	0	0	0	
18	006A9198-1BB4	1	0	0	0	0	0	0	0	1	0	0	0	
19	006BF25A-315A	0	0	1	0	0	0	0	0	0	0	0	0	
20	006E3A23-F786	2	0	8	1	0	0	0	1	1	0	0	0	
21	006F8D62-651E	0	0	0	0	0	1	0	0	0	0	0	0	
22	0070A69E-281C	0	0	1	0	1	0	0	0	0	0	0	0	
23	0073C602-D9AE	0	0	2	0	0	0	0	0	0	0	0	0	
24	007CB9B1-F4DI	1	0	0	0	1	0	0	1	1	1	0	0	
25	008383B8-BF24	1	0	2	0	0	0	0	4	0	0	0	0	

- Transcript table



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	PatientGuid	Height_Max	Weight_Max	BMI_Max	SystolicBP_Max	DiastolicBP_Ma	RespiratoryRate	Temperature_Ma	Height_Min	Weight_Min	BMI_Min	SystolicBP_Min	DiastolicBP_Min	RespiratoryRate T
2	00023761-9D8D	68	185	28.972	134	86	16.91502579	97.9	67	182	27.67	120	62	16
3	0005D9BD-0247	67	137	22.796	152	82	16.91502579	98.6	65	0	0	125	59	15
4	000B4862-7CE7	65.89672602	100.09	16.654	126.728523	76.45129703	20	97.76533086	65	0	0	80	56	14
5	00110ABC-DAB	71.5	215	29.983	130	80	16.91502579	97.76533086	65.89672602	0	0	125	75	12
6	002667F4-B9A3	65.89672602	109	22.013	160	90	20	97.76533086	59	0	0	112	70	16.91502579
7	0029BBC8-7C22	68	186	28.278	163	98	16.91502579	99.8	65.89672602	0	0	101	70	16.91502579
8	003CEE1F-5BF5	66	251.327	22.75	160	82	14	97.76533086	60	105	0	116	54	10
9	004382BD-E31F	63	242	42.864	152	92	16.91502579	97.76533086	63	230	40.738	134	80	16.91502579
10	004AACFF-5D6	67	180	26.623	128	96	16.91502579	98.6	65.89672602	0	0	102	80	16
11	004CC712-BAD	68.5	161.7	24.226	126.728523	78	18	97.76533086	65.89672602	0	0	115	72	16.91502579
12	004EA6A7-6159	67	142	22.238	126.728523	76.45129703	16.91502579	98.3	65.89672602	0	0	96	64	16.91502579
13	005F61C9-E537	65.89672602	180	32.341	152	98	20	98.9	61.5	0	0	110	60	15
14	00607D4B-5F8F	65.89672602	155	22.629	126.728523	76.45129703	16.91502579	97.76533086	65	0	0	84	52	16.91502579
15	0063B34F-C2C4	68	290	45.415	133	85	20	98.7	65.89672602	0	0	106	72	18
16	0066553F-732C	65.89672602	265	46.938	148	88	16.91502579	97.76533086	63	0	0	110	72	16.91502579
17	006948F2-1118-	67.5	170	27.436	130	82	18	102.6	66	164	26.23	110	66	14
18	006A9198-1BB4	65.89672602	96	18.747	126.728523	82	16.91502579	99.1	60	0	0	98	62	16.91502579
19	006BF25A-315A	65.89672602	155	28.347	137	81	16.91502579	97.76533086	62	0	0	116	66	16.91502579
20	006E3A23-F786	72.5	241	32.233	157	79	18	98.4	65.89672602	221	0	118	52	10
21	006F8D62-651E	67	155	24.481	139	76.45129703	16.91502579	97.76533086	65.89672602	0	0	126.728523	68	16.91502579
22	0070A69E-281C	65.89672602	218.2	38.648	138	88	16.91502579	97.76533086	62	0	0	118	76.45129703	14
23	0073C602-D9AE	66	154	24.854	126.728523	85	16.91502579	97.76533086	65.89672602	0	0	103	76.45129703	16.91502579
24	007CB9B1-F4DI	72	218	29.563	202	87	22	98	72	190	25.766	130	78	20
25	008383B8-BF24	68	300	46.982	150	90	22	98.6	65.89672602	0	0	110	76.45129703	16

- physician speciality



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	PatientGuid	PhySp_Academ	PhySp_Addictior	PhySp_Adolesc	PhySp_Adult_R	PhySp_Allergy_I	PhySp_Alternati	PhySp_Anesthe	PhySp_Cardiova	PhySp_Chiropra	PhySp_Clinical_	PhySp_Clinical_	PhySp_Critical_	PhySp_Dermato
2	00023761-9D8D	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0005D9BD-0247	0	0	0	0	0	0	0	0	0	0	0	0	0
4	000B4862-7CE7	0	0	0	0	0	0	0	0	0	0	0	0	0
5	00110ABC-DAB	0	0	0	0	0	0	0	0	0	0	0	0	0
6	002667F4-B9A3	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0029BBC8-7C22	0	0	0	0	0	0	0	0	0	0	0	0	0
8	003CEE1F-5BF5	0	0	0	0	0	0	0	0	0	0	0	0	0
9	004382BD-E31F	0	0	0	0	0	0	0	0	0	0	0	0	0
10	004AACFF-5D6	0	0	0	0	0	0	0	0	0	0	0	0	0
11	004CC712-BAD	0	0	0	0	0	0	0	0	0	0	0	0	0
12	004EA6A7-6159	0	0	0	0	0	0	0	0	0	0	0	0	0
13	005F61C9-E537	0	0	0	0	0	0	0	0	0	0	0	0	0
14	00607D4B-5F8F	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0063B34F-C2C4	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0066553F-732C	0	0	0	0	0	0	0	0	0	0	0	0	0
17	006948F2-1118-	0	0	0	0	0	0	0	0	0	0	0	0	0
18	006A9198-1BB4	0	0	0	0	0	0	0	0	0	0	0	0	0
19	006BF25A-315A	0	0	0	0	0	0	0	0	0	0	0	0	0
20	006E3A23-F786	0	0	0	0	0	0	0	0	0	0	0	0	0
21	006F8D62-651E	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0070A69E-281C	0	0	0	0	1	0	0	0	0	0	0	0	0
23	0073C602-D9AE	0	0	0	0	0	0	0	0	0	0	0	0	0
24	007CB9B1-F4DI	0	0	0	0	0	0	0	0	0	0	0	0	0
25	008383B8-BF24	0	0	0	0	0	0	0	0	0	0	0	0	0

2.1.2 Input schema

Feature name	Datatype	Size	Null/Required
Age	int	3	Required

2.2 Predicting Disease

- The system displays the choices of the disease.
- The User chooses the target disease by clicking one of the available diseases.
- The User selects the disease.
- The system presents the set of inputs required from the user.
- The user gives required information.
- The system should be able to predict whether infected for the chosen disease based on the user information.

2.3 Logging

- We should be able to log every activity done by the user.

- The System identifies at what step logging required
- The System should be able to log each and every system flow.
- Developers can choose logging methods. You can choose database logging/ File logging as well.
- System should not be hung even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

2.4 Database

System needs to store every request into the database and we need to store it in such a way that it is easy to retrain the model as well.

- The User chooses the disease.
- The User gives required information.
- The system stores each and every data given by the user or received on request to the database. Database you can choose your own choice whether MongoDB/ MySQL.

2.5 Deployment

- AWS

3 Technology stack

Front End	HTML/CSS/JS/React
Backend	Python Django
Database	MongoDB/MySql
Deployment	AWS

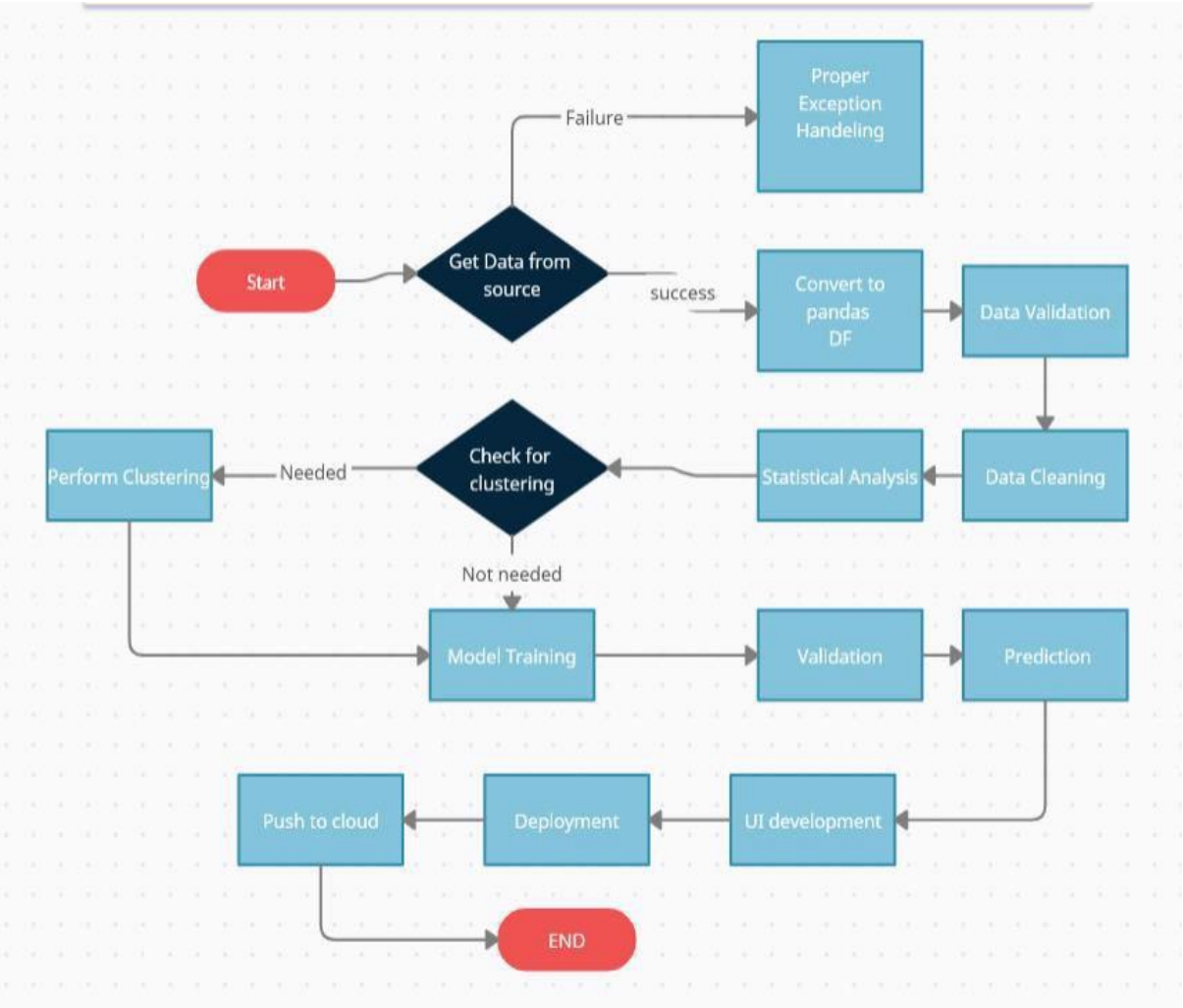
4 Proposed Solution

Refer: https://arxiv.org/abs/1808.04928

Based on the actual research paper, if we are using history of the patient to predict the future, then we might want to consider using LSTM. However, drawing a baseline in the form of some Machine Learning algorithm would be helpful. Why making a baseline model important? Well, to compare the performance of our actual model, let say LSTM in this case, is very important to ascertain that we are in the right direction as if performance of LSTM is not better than the baseline model then there is no point of using LSTM.

- Baseline Model: Logistic Regression, since this is a classification problem.
- Actual model: LSTM.

5 Model training/validation workflow



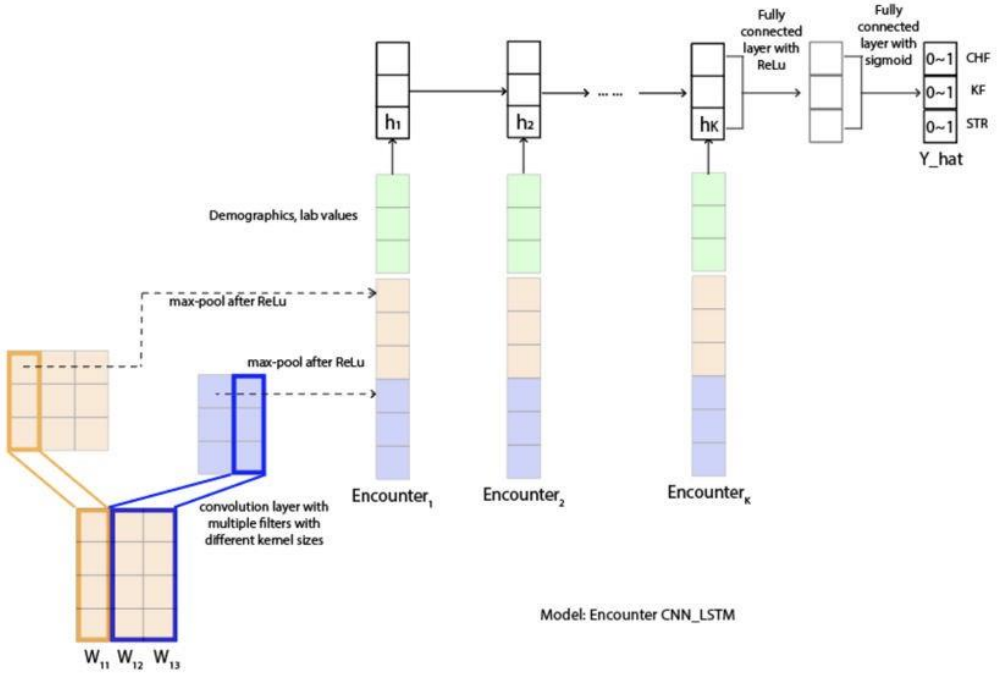
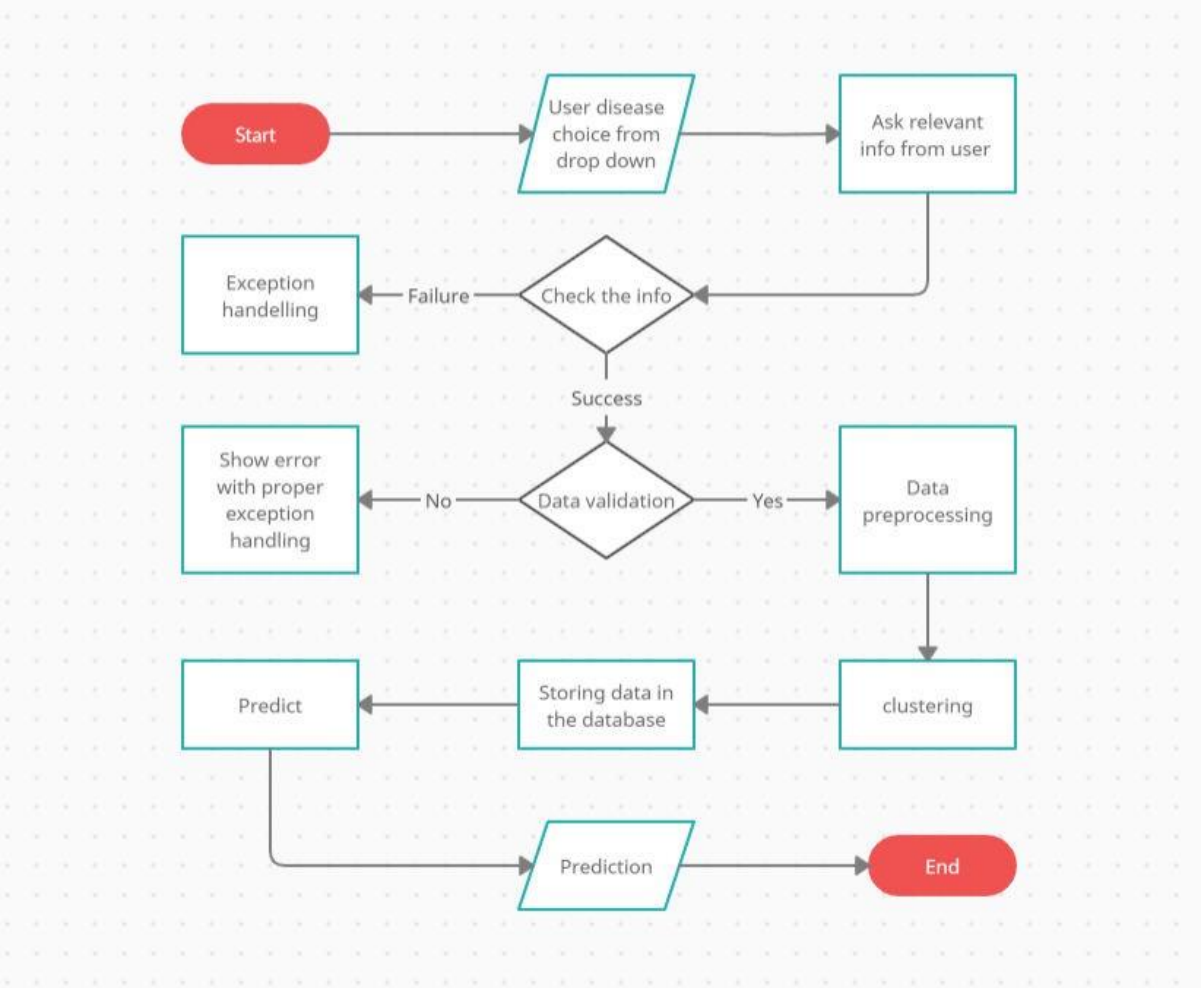


Figure 5: Illustration of the encounter-level hierarchy model: each encounter is encoded into a hidden vector via a CNN. Lab values and demographics are concatenated to the CNN output, then fed into an LSTM to generate prediction.

6 User I/O workflow



7 Exceptional scenarios

Step	Exception	Mitigation	Module

8 Test cases

Test case	Steps to perform test case	Module	Pass/Fail

9 Key performance indicators (KPI)

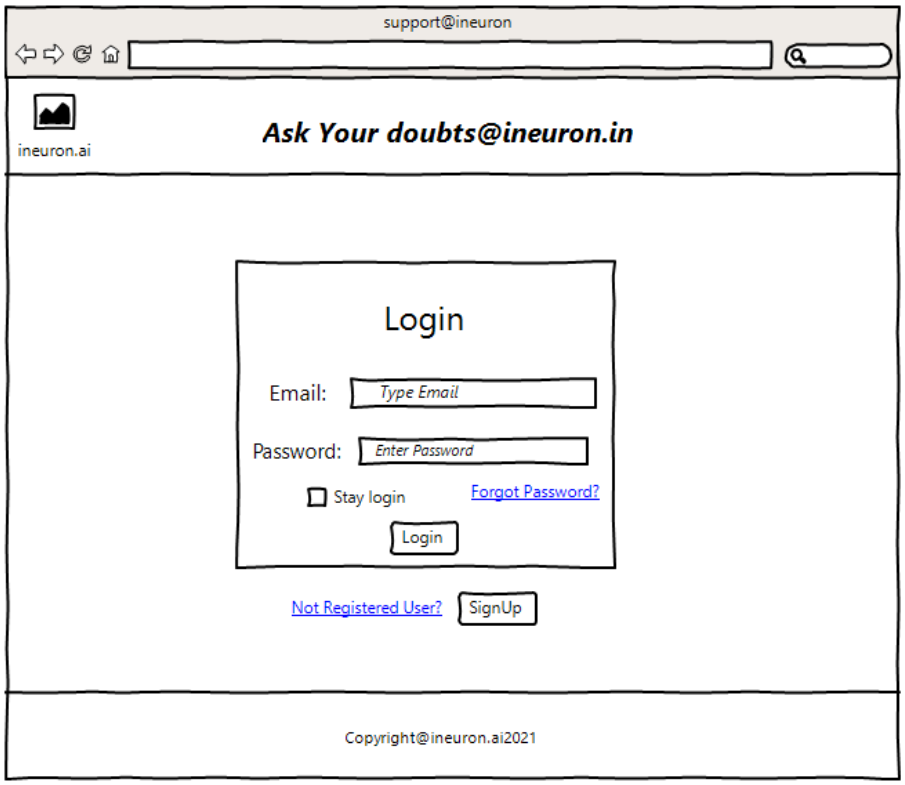
- Time and workload reduction using the EHR model.
- Comparison of accuracy of model prediction and doctor's prediction.
- Number of times a patient visits the hospital.
- Time between symptom onset and detection of illness/visit to hospital.
- Immunity of the patient (based on previous illnesses).
- Vaccines the patient has taken.
- Length of stays in the hospital.

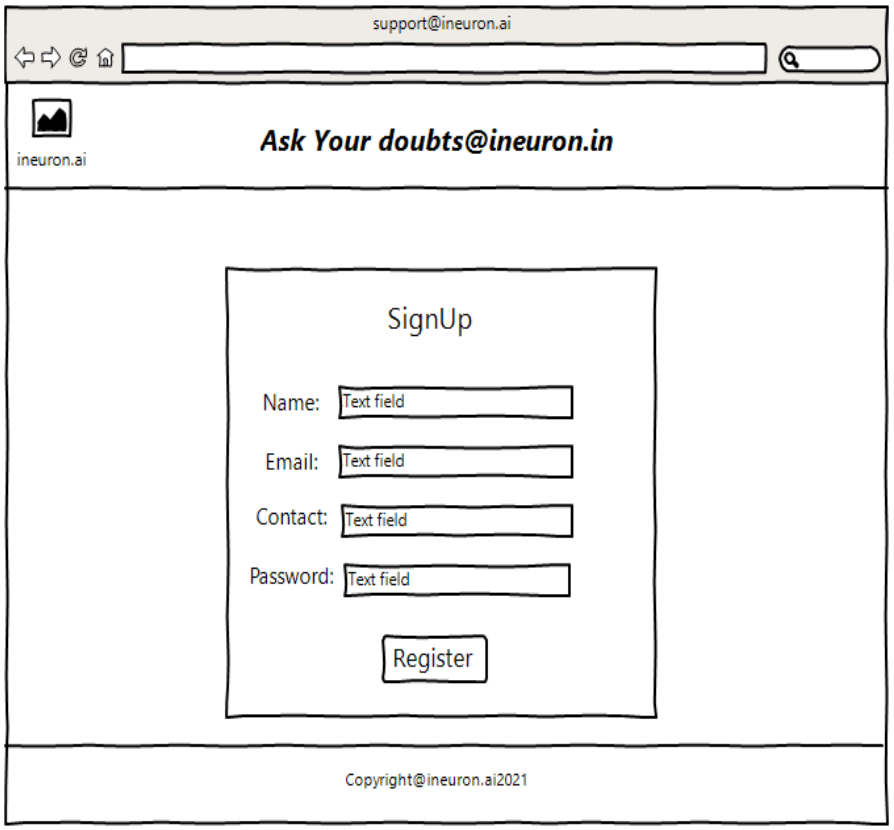
WIREFRAME DOCUMENT

Automatically Assign Tickets in CRM Tools to a Particular Team

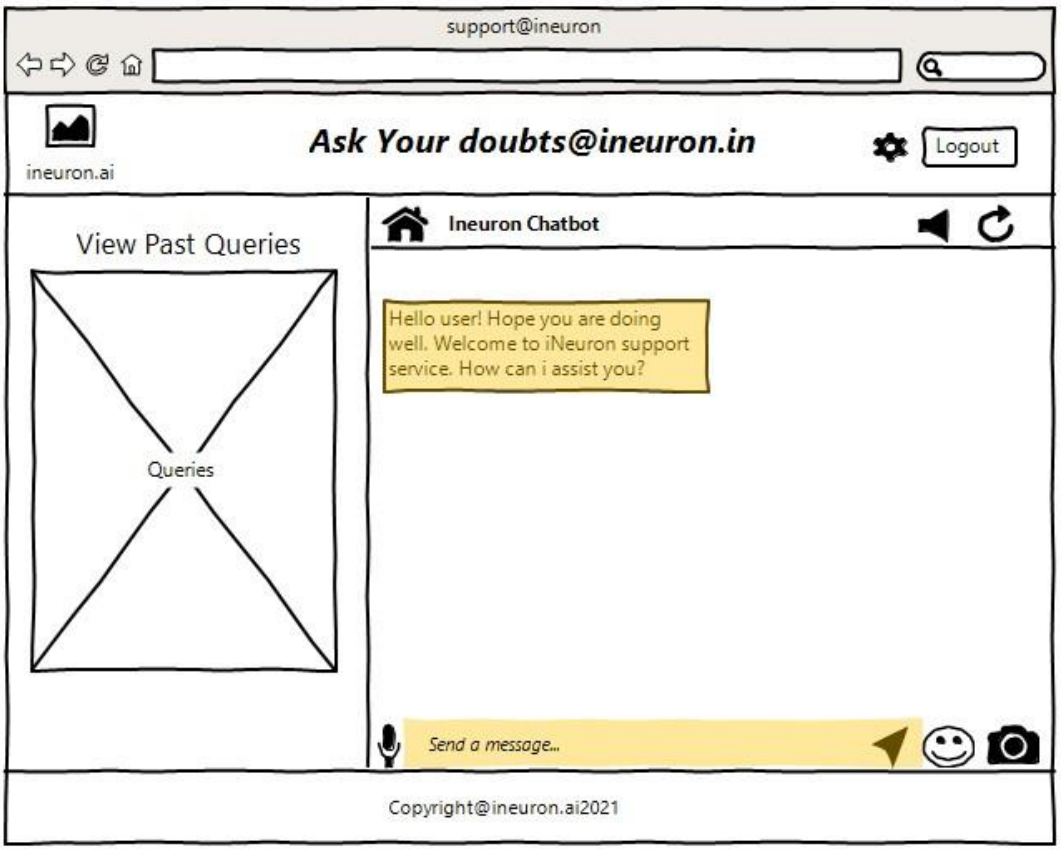
Team Lead:
Created by: Asif Khan
Batch: 10th October

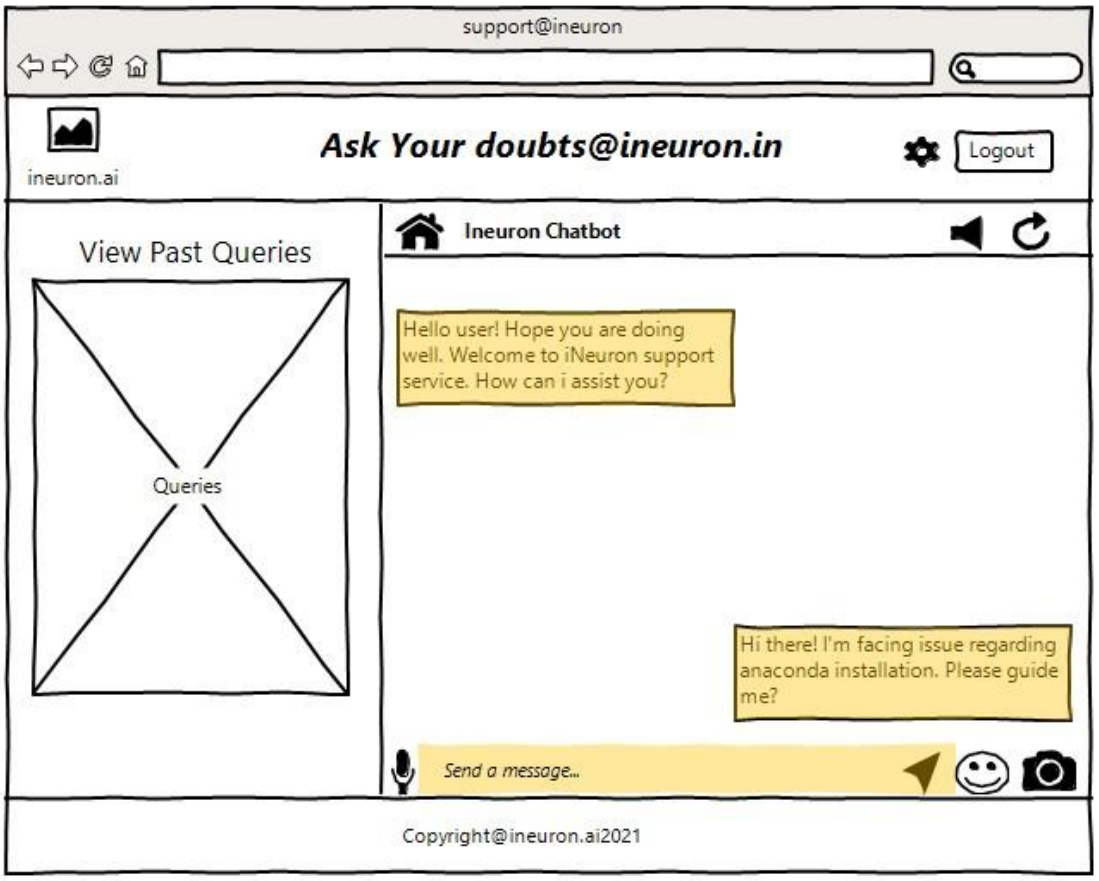
Patient Panel

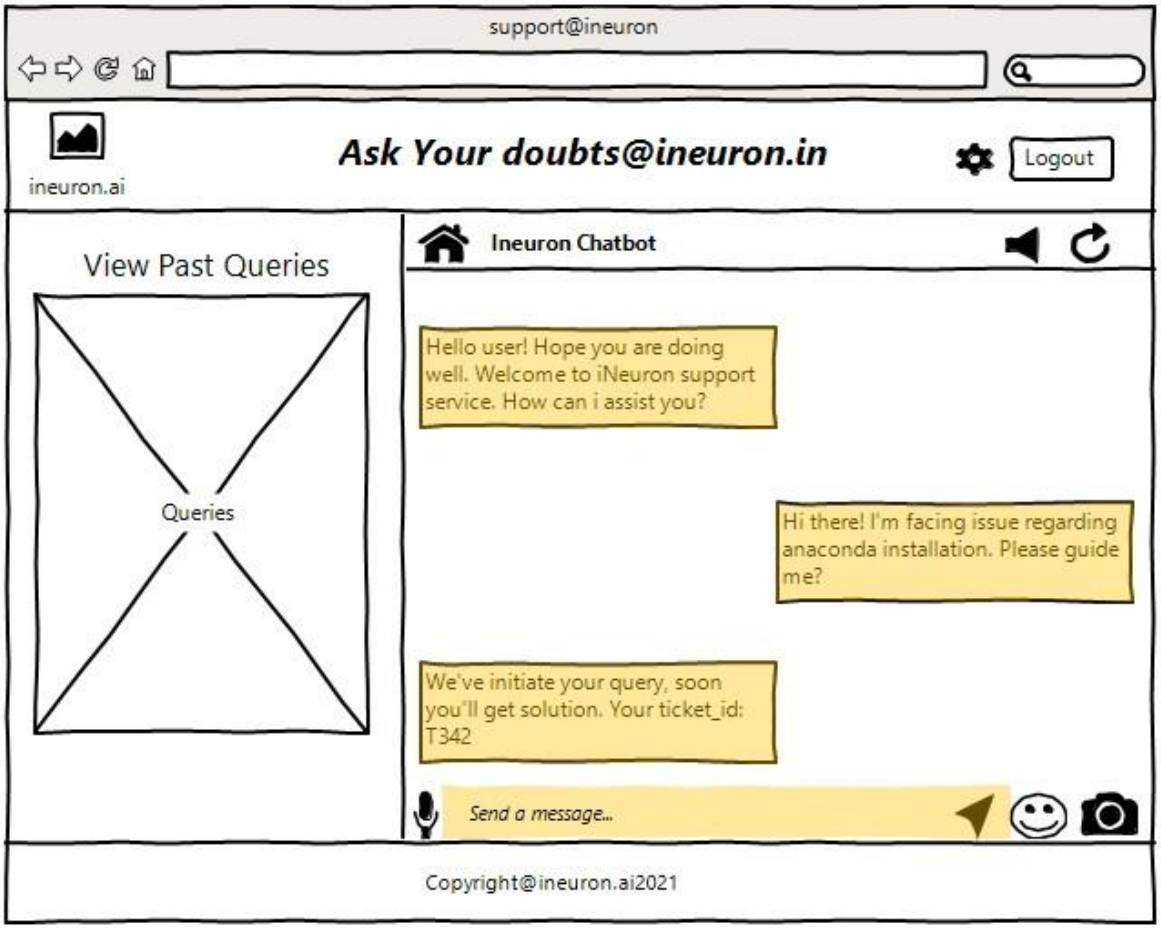




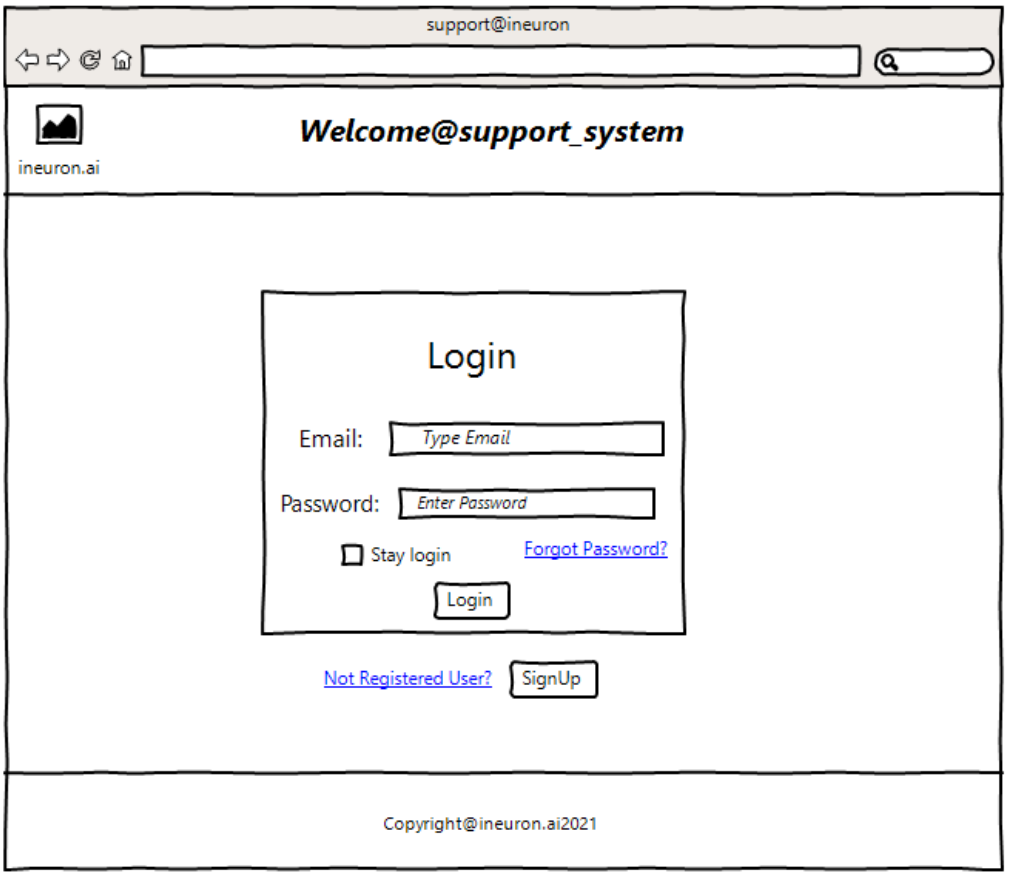
Patient Dashboard (Chatbot)

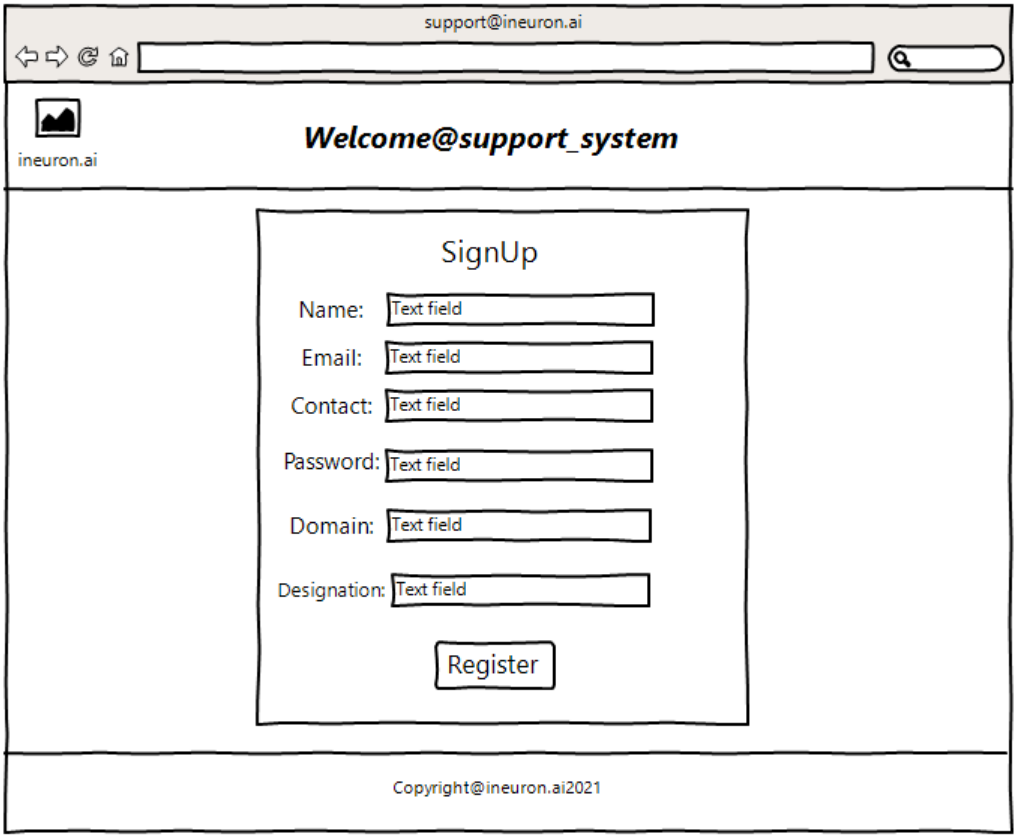



support@ineuron
ineuron.ai
Ask Your doubts@ineuron.in
Logout
View Past Queries
Queries
Ineuron Chatbot
Hello user! Hope you are doing well. Welcome to iNeuron support service. How can i assist you?
Hi there! I'm facing issue regarding anaconda installation. Please guide me?
Send a message...
Copyright@ineuron.ai2021


support@ineuron
ineuron.ai
Ask Your doubts@ineuron.in
Logout
View Past Queries
Queries
Ineuron Chatbot
Hello user! Hope you are doing well. Welcome to iNeuron support service. How can i assist you?
Hi there! I'm facing issue regarding anaconda installation. Please guide me?
We've initiate your query, soon you'll get solution. Your ticket_id: T342
Send a message...
Copyright@ineuron.ai2021

Backend (Admin Panel)





Supervisor Dashboard

The Signup and Login panel in above page uses by whole backend panel i.e. all expert team members and by supervisor.