Low Level Design (LLD)

UGV (Unmanned Ground Vehicle) based Surveillance

Revision Number: 1.8

Last date of revision: 02/06/2021

Khushali Shah

Amit K Gupta

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 18th May 2020 | 1.1 | First Draft | Amit K Gupta |
| 20th May 2020 | 1.2 | Added Workflow chart | Amit K Gupta |
| 20th May 2020 | 1.3 | Added Exception Scenarios Overall, Constraints | Vrunda Patel |
| 21st May 2021 | 1.4 | Added KPIs | Sukritha Joshi |
| 26th May 2021 | 1.5 | Added user I/O flowchart | Amit K Gupta |
| 26th May 2021 | 1.6 | Added EHR, LSTM model diagrams | Nagesh |
| 31st May 2021 | 1.7 | Added dataset overview and updated user I/O flowchart. | Amit K Gupta |
| 02nd June 2021 | 1.8 | Restructure and reformat LLD | Khushali Shah |
|  |  |  |  |

**Contents**

[Document Version Control 2](#_gjdgxs)

[**Abstract** 4](#_30j0zll)

[1](#_1fob9te) Introduction 5

[1.1](#_3znysh7) Why this Low-Level Design Document? 5

[1.2](#_2et92p0) Scope 6

[1.3](#_tyjcwt) Constraints 6

[1.4](#_3dy6vkm) 1.3 Risks 6

[1.5](#_1t3h5sf) 1.4 Out of Scope 7

[2](#_4d34og8) Technical specifications 7

[2.1 Predicting Disease 7](#_2s8eyo1)

[2.2 Logging 7](#_17dp8vu)

[2.3 Database 7](#_3rdcrjn)

[3](#_26in1rg) Technology stack 8

[4](#_lnxbz9) Proposed Solution 9

[5](#_35nkun2) Model training/validation workflow 10

[6](#_1ksv4uv) User I/O workflow 12

[7](#_44sinio) Exceptional scenarios 13

**Abstract**

With high increase in vehicles on the road, problem of traffic congestion and accidents has increased substantially. To overcome these problems, the in-depth analysis of causes such as number of traffic rules followed, is required. Therefore, continuous monitoring of traffic on highways and huge roads is mandatory. An Automatic Traffic Control System can prove to be a solution to above mentioned problems.

Automatic Traffic Counter Control can also help in drawing inferences from the recorded data.

# Introduction

## 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.

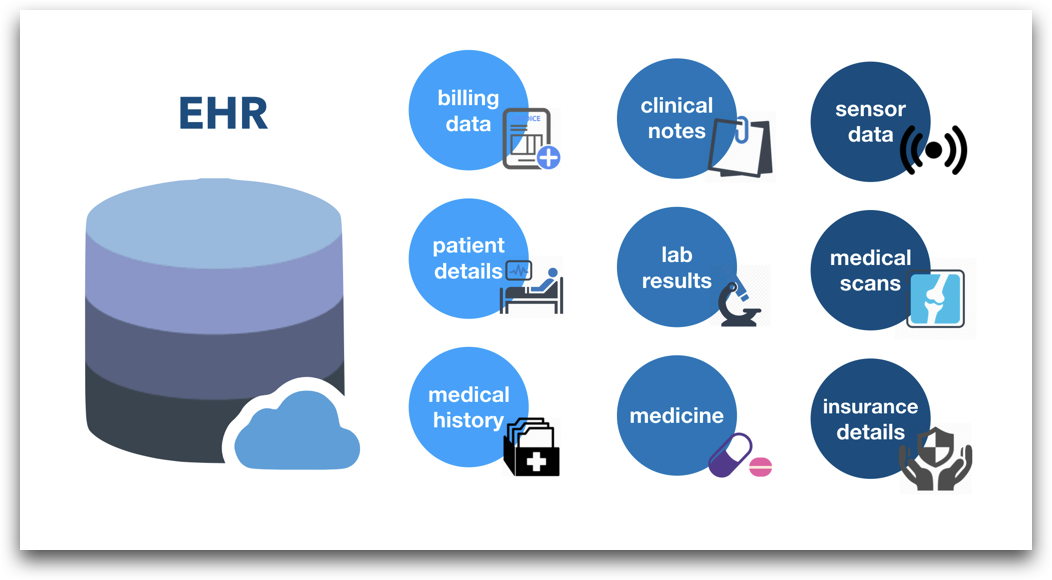
The main objective of the project is to predict if a person can get a chronic disease in his/her future based on the EHR. EHR stands for Electronic Health Record, EHR is nothing but a dataset of medical history of the patients.

EHRs are a vital part of health IT and can:

* Contain a patient’s medical history, diagnoses, medications, treatment plans, immunization dates, allergies, radiology images, and laboratory and test results
* Allow access to evidence-based tools that providers can use to make decisions about a patient’s care
* Automate and streamline provider workflow

An [electronic health record](https://www.healthit.gov/providers-professionals/learn-ehr-basics) (EHR) contains patient health information, such as:

* Patient demographics
* Progress notes
* Vital signs
* Medical histories
* Diagnoses
* Medications
* Immunization dates
* Allergies
* Radiology images
* Lab and test results



This project shall be delivered in two phases:

Phase 1: All the functionalities with PyPi packages.

Phase2: Integration of UI to all the functionalities.

## Scope

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

## Constraints

We will only be selecting a few of the chronic diseases.

## Risks

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

## Out of Scope

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

# Technical specifications

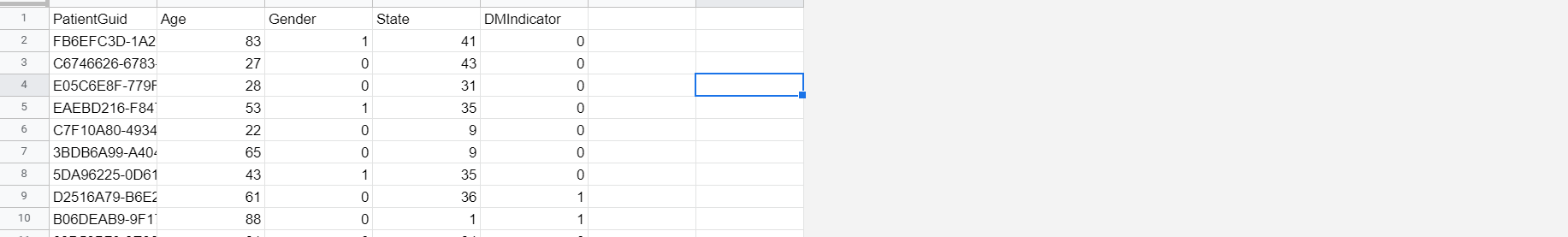
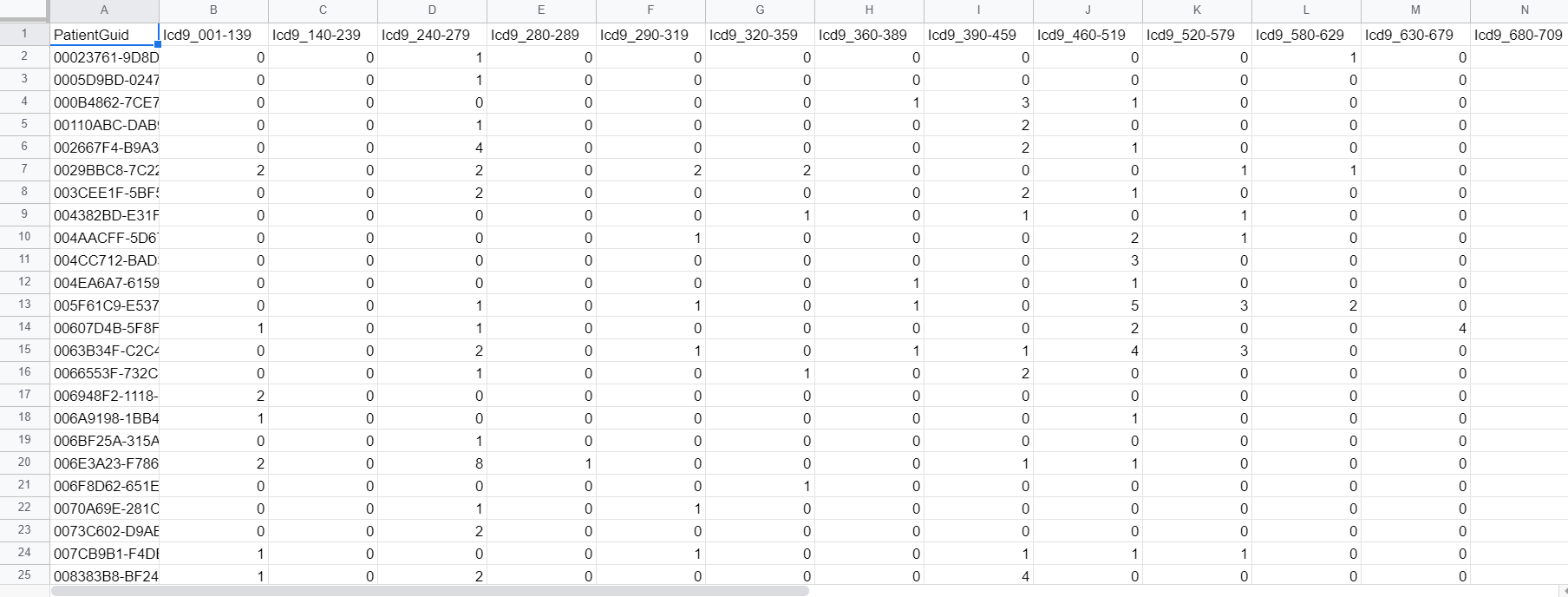
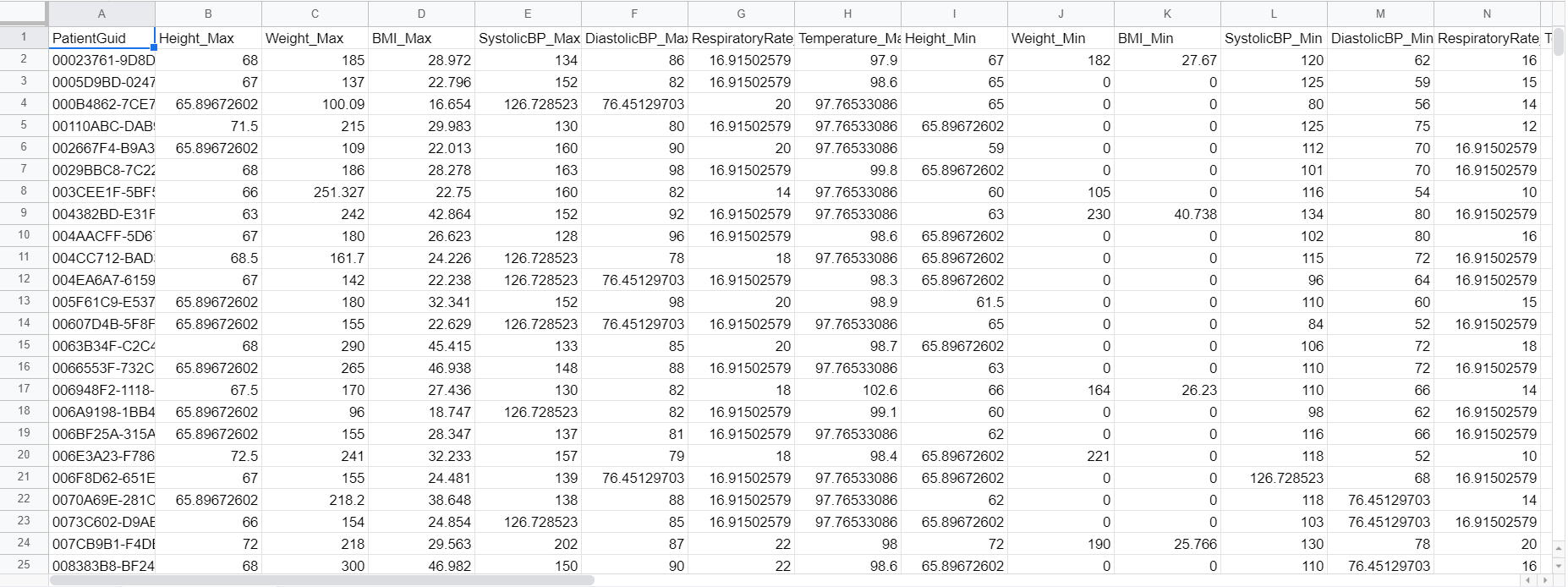
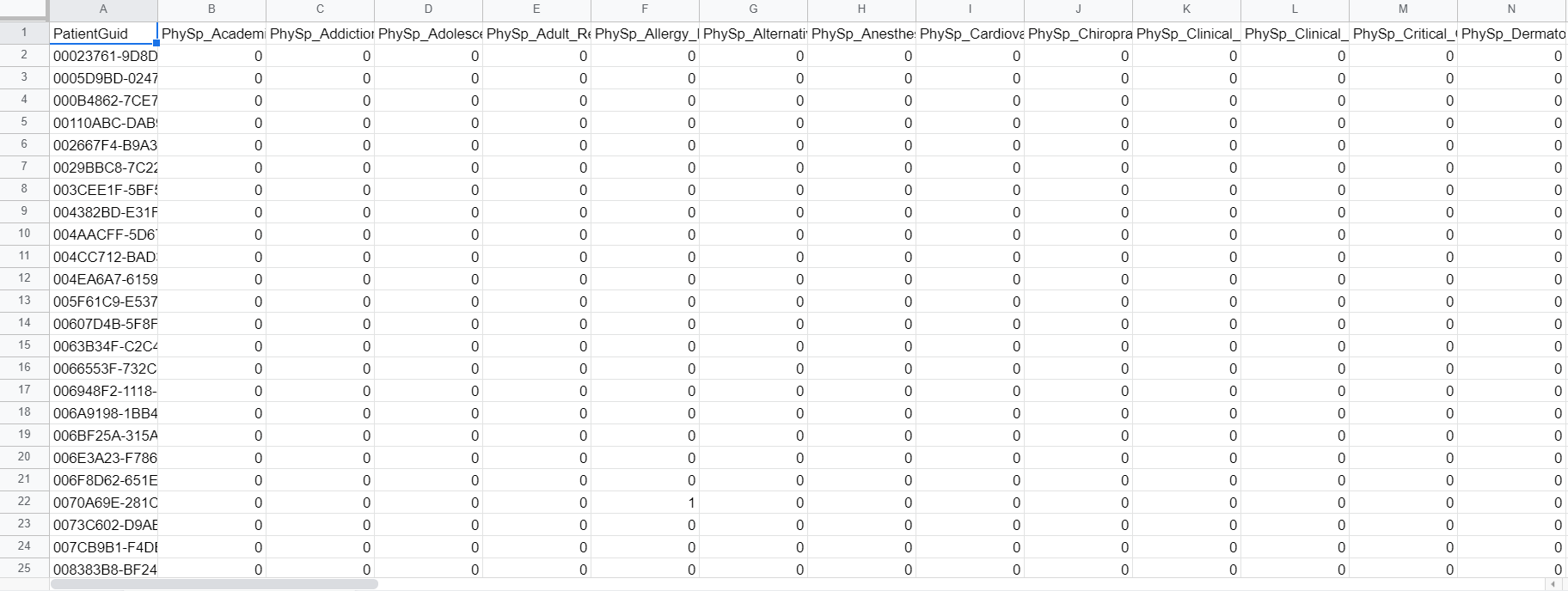
## 2.1 Dataset

|  |  |  |
| --- | --- | --- |
| **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
* Diagnosis table
* Transcrip table
* physician speciality

## 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.

1. The User chooses the disease.

2. The User gives required information.

3. 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**

1. AWS



# Technology stack

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

# Proposed Solution

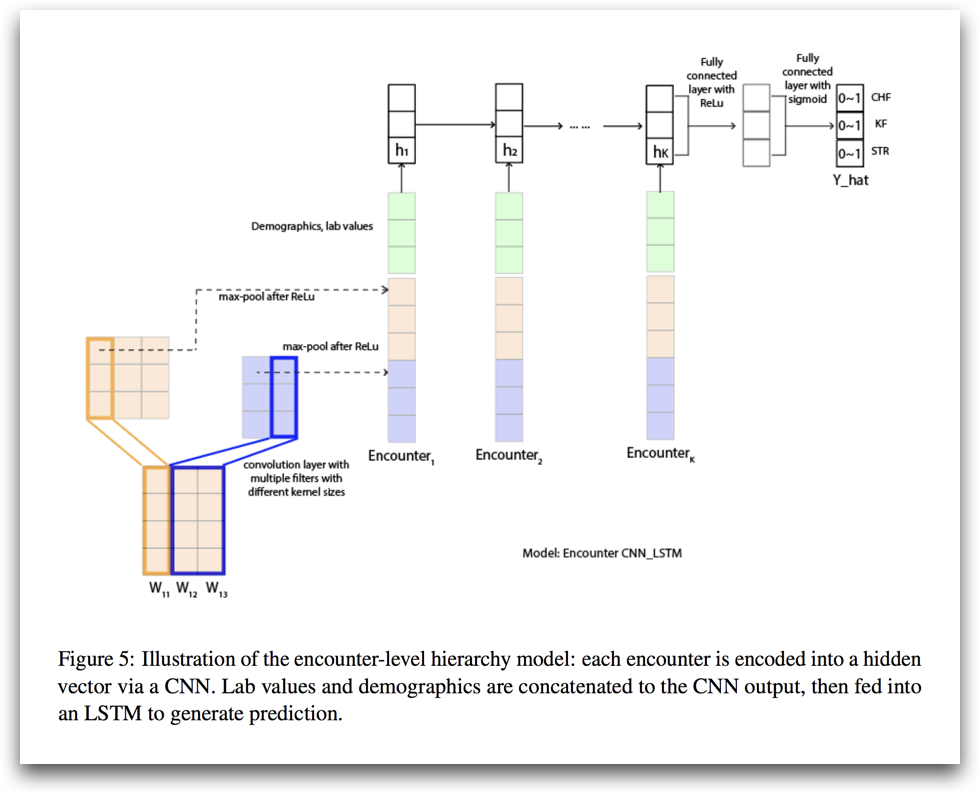
refer: <https://arxiv.org/abs/1808.04928>

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.

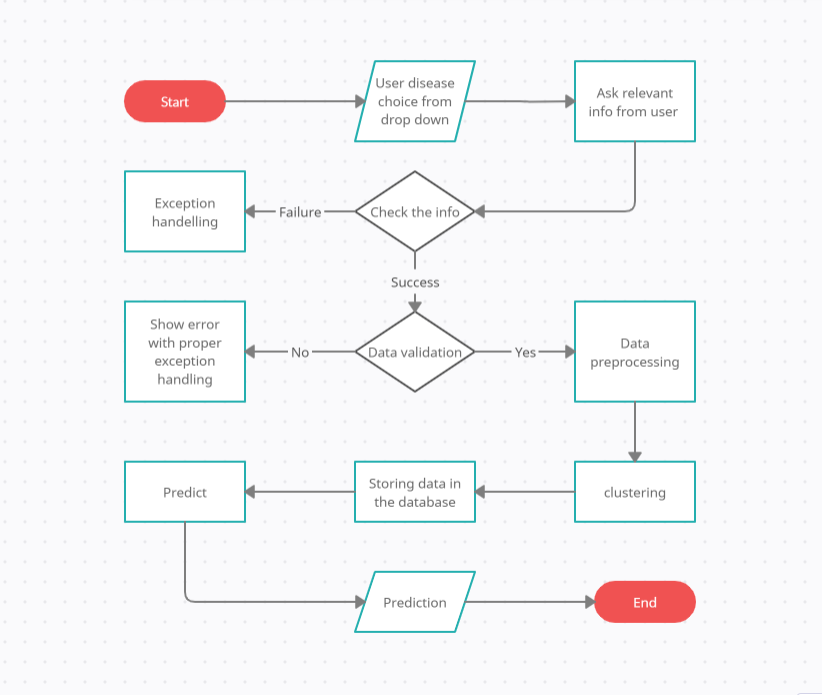
1. Baseline Model: Logistic Regression, since this is a classification problem.
2. Actual model: LSTMs.

# Model training/validation workflow



****

# User I/O workflow

****

# Exceptional scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Exception | Mitigation | Module |
| 18th May 2020 | 1.1 | First Draft | Amit K Gupta |
| 20th May 2020 | 1.2 | Added Workflow chart | Amit K Gupta |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Steps to perform test case | Module | Pass/Fail |
|  |  |  |  |

# 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 patient (based on previous illnesses).
* Vaccines the patient has taken.
* Length of stays in hospital.