Low Level Design (LLD)

FTD (Fraud Transaction Detection)

Revision Number: 1.8

Last date of revision: 25/03/2023

Shaik mahammad rafi

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 10th feb 2023 | 1.1 | First Draft | Shaik mahammad rafi |
| 17th feb 2023 | 1.2 | Added Workflow chart | Shaik mahammad rafi |
| 20th feb 2023 | 1.3 | Added Exception Scenarios Overall, Constraints | Shaik mahammad rafi |
| 20th Mar 2023 | 1.4 | Deployement | Shaik mahammad rafi |
| 21th mar 2021 | 1.5 | Added user I/O flowchart | Shaik mahammad rafi |
| 22th mar 2021 | 1.6 | Added EHR, LSTM model diagrams | Shaik mahammad rafi |
| 23th mar 2021 | 1.7 | Added dataset overview and updated user I/O flowchart. | Shaik mahammad rafi |
| 24 mar 2021 | 1.8 | Restructure and reformat LLD | Shaik mahammad rafi |
|  |  |  |  |

**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**

Fraud detection is a set of activities undertaken to prevent money or property from

being obtained through false pretenses. Fraud detection is applied to many industries

such as banking or insurance. In banking, fraud may include forging checks or using

stolen credit cards. Other forms of fraud may involve exaggerating losses or causing an

accident with the sole intent for the pay-out. With an unlimited and rising number of

ways someone can commit fraud, detection can be difficult to accomplish. Fraud

detection is a critical issue for retailers determined to prevent losses and preserve

customer trust.

With the increase in digitalization, there is also increase in the fraudulent activities

happening in various domains, mainly in the retail domain. These are detrimental to the

ecosystem of online transactions. Machine learning provides an intelligent option in

dealing with this challenge.

# 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 completed transaction is Correct or Fraud using this Machine learning model . FTD stands for Fraud Transaction detection , FTD is nothing but a dataset of history of the Transactions (online).

Fraud Transaction Detection data:

* Contain a customer history like customer id number customer name customer pincode etc
* It also contain terminal feature like transaction dataetime , transaction amount ..etc.

The Fraud transaction data contains features such as:

* TX\_DATETIME
* CUSTOMER\_ID
* TERMINAL\_ID
* TX\_AMOUNT
* TX\_TIME\_SECONDS
* TX\_TIME\_DAYS
* TX\_FRAUD
* TX\_FRAUD\_SCENARIO

This project shall be delivered in two phases:

Phase 1: All the functionalities with PyPi packages.

Phase2: Deployment in AWS

## Scope

This software system will be a Web application This system will be designed to detect the fraud transaction, improve healthy relation ship between online customers and retailers, It is also useful in Banking domain and many other money transactions domains to detect fraud transactions and Remove them.. This system is designed to predict the Fraud transactions from transaction history information such as tx\_amount, tx\_time, tx\_time\_seconds, and customer\_id ..etc.

## Constraints

We will only be selecting a few of the fraud \_transactions.

## 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** |
| Fraud Transaction | yes | https://github.com/Fraud-Detection-Handbook/simulated-data-raw. |
| Fraud Transaction | No |  |

## 2.1.1 dataset overview

Consists of a table like shown below,These table consist of table consists of customer id terminal id, tx\_date\_time, Tx\_amount ,Tx-time..etc and most importantly we have the randomly generated data of transactions

The dataset has very large data set 17,00,000 instances so we only taking 4,00,000 Instances to build a model

There are a total of 3,00,000 transactions in the training set 1,00,000 in the test set.

## 2.1.2 Input schema

|  |  |  |
| --- | --- | --- |
| **Feature name** | **Datatype** | **Null/Required** |
| TX\_DATE\_TIME | Datetime64 | Required |
| CUSTOMER\_ID | int | required |
| TERMINAL\_ID | int | required |
| TX\_AMOUNT | float | required |
| TX\_TIME\_SECONDS | int | required |
| TX\_TIME\_DAYS | int | required |
| TX\_FRAUD\_SCENARIO | int | required |

## 2.2 Prediction

* Deployed model in AWS.Then the deployed machine will give an link.
* The link is used to predict transaction.
* By clicking on the link it will give an UI , where we can insert our file and then getting predictions.

## 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



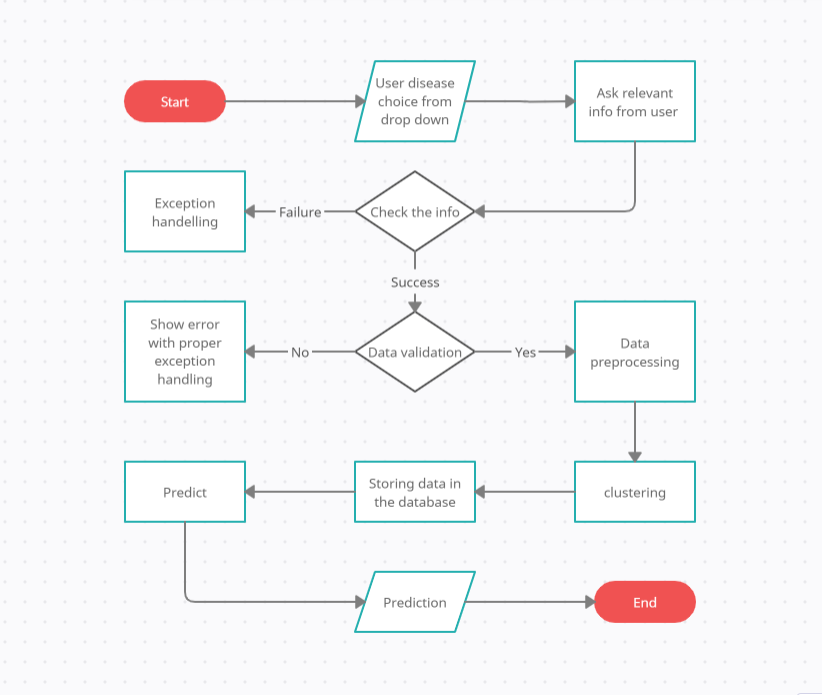
# Technology stack

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

# Model training/validation workflow



# User I/O workflow

****

## 5.DEPLOYEMENT IN AWS

1. 1.After deploying in aws we can send a file which contains information about the transactions and detecting them they are correct or fraud.

2.After completing in AWS the deployment machine will give a link for accessing the model. using these link we can predict transactions. In any machine.