Comprehensive Design and Analysis Project

Design Document

IT16120280 – N.P. Seneviratne

Fraud Detection Based on Transaction Data

Bachelor of Science (Honors) in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

May 2019

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this Design Document is to present a detailed description of "fraud detection based on transaction data" which will detect a fraud as early as possible in a banking environment using transaction data. Further, this will elaborate the functional and non-functional requirements of the system. And, this document will depict the software and hardware interfaces which these modules use and how those will be connected to give the user a great experience. This document will have a design architecture of the system, which will precisely divide the system into specific components and it will show the connections which will pass data to communicate through the system.

1.2 Scope

This document includes an abstract of the module – "fraud detection based on transaction details". It will include high level information that is required to understand the major contributions of this module to the larger system – "fraud detection based on human behavioral patterns". What this module is supposed to do, how it will contribute to the whole system, what data communication methods will be used, and most importantly, the machine learning aspects of the module will be discussed. Furthermore, this document will introduce you some wireframes regarding to the user interface of this module.

1.3 Definitions, Acronyms, and Abbreviations

UI – User Interface

LAN – Local Area Network

ML – Machine Learning

POS – Point Of Service

ATM – Automatic Teller Machine

1.4 Overview

This fraud detection module which is based on transaction data will detect a fraud as early as possible. When a bank customer does a transaction, this module will be activated. If this user is registered on the system, they will be analyzed, else the user will be registered to the system as a new user. Then the transaction data will be compared to the learned data of the system. If the current transaction is likely a fraud, it will be flagged and analyzed in deep. The flagged transaction will be alerted immediately, and the personnel should take the necessary actions afterwards.

2 Overall Descriptions

This software will enable the privileged persons of the bank to be logged in. The user will be able to see a dashboard after logging in. In the transaction-based fraud detection model UI page, the user will see a combo box to select which counter or teller machine. After selecting one, currently investigating transaction details, all the transaction details evaluated in this counter / teller will be shown. They can be sorted according to user's criteria.

While the user sees this, the underlying machine learning algorithm will run investigating the transactions. According to learned behaviors, patterns it will predict a possibility of a fraudulent transaction. Machine learning algorithms can identify patterns that would not be visible to the human eye. When a most likely fraudulent transaction is found, an alert will be popped up in the required personnel i.e. bank manager, counter personnel, security personnel.

All the transaction details and predicted details will be stored in a secured database.

2.1 Product perspective

Riskified is a fraud prevention, competitive application that is in the current market. They have a ecosystem which consists of billions of transaction recorded. A lot of information will be recorded such as names, emails, shipping, billing addresses. For each transaction, they will run an elastic query to get a list of similar historical orders. This will help to categorize legitimate and fraud transactions.

The above stated functionality of *Riskify* will be implemented in this research's transaction module too. All the historical data will be used for learning purposes of the machine learning model.

2.1.1 System interfaces

The windows application installed in the administrator's personal computer.

The database which stores the transaction information of the counters and the teller machines.

The database to store all the predicted values of the transactions.

2.1.2 User interfaces

Below are the sketches of user interfaces used by this software module. They will be described in details in section 3.

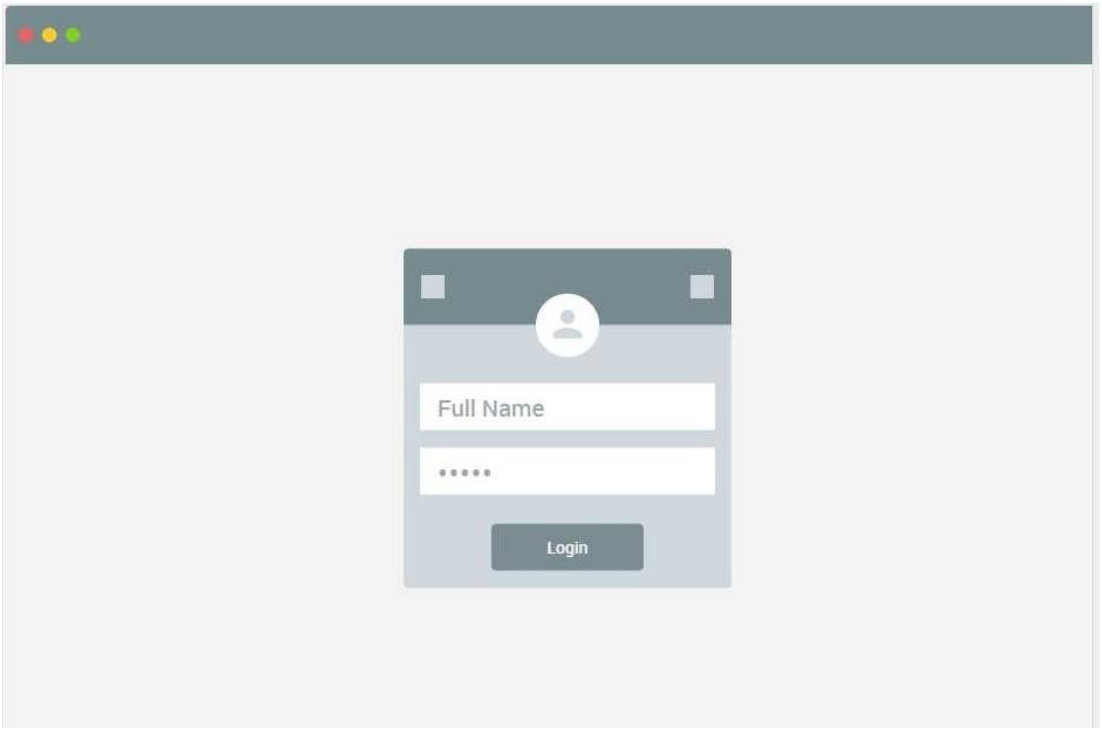


Figure: Login Page

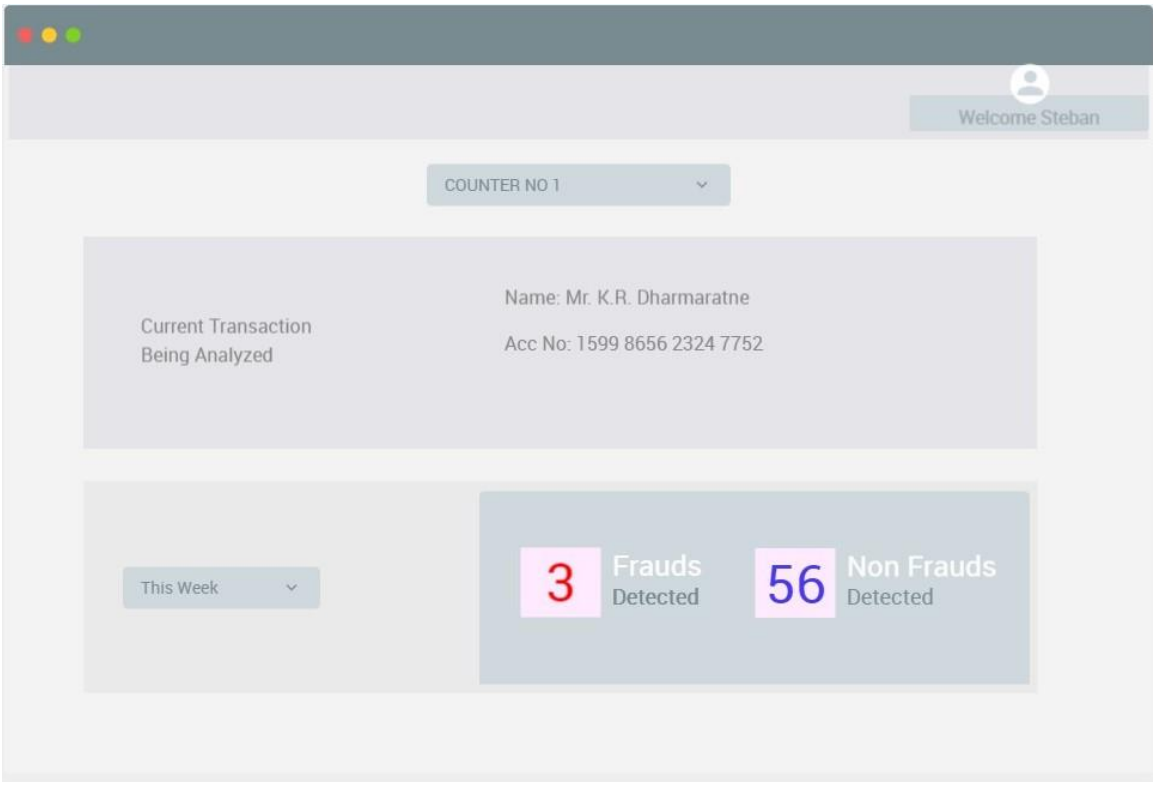


Figure: Transaction Analyzing Dashboard

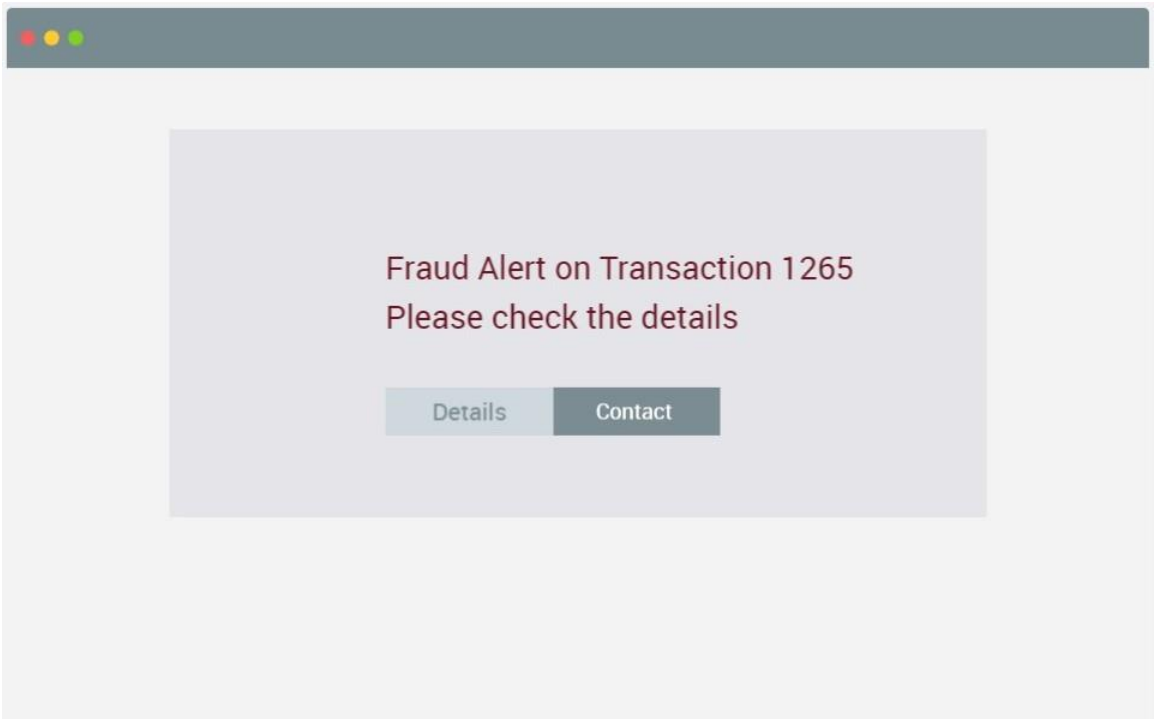


Figure: Alert Popup

2.1.3 Hardware interfaces

Specific hardware devices will not be needed for this module of the system since it is based on analyzing transaction details. But a functioning system unit is required to use the deploy the system and to monitor it. Other modules of this system will need specific hardware devices to detect and analyze human behavioral patterns to identify the fraud patterns.

2.1.4 Software interfaces

The machine learning models will be trained by using TensorFlow framework. And the web application (UI) will be developed using the Angular framework. Since both frameworks are google products and since they are tailored to be used together, it is a good combination. The customer information will be directly communicated to the TensorFlow model, while the dashboard users (personnel who control the software) would access the web application UI panel from the Angular driven dashboard.

For customer tracking purposes, we may need a small scaled data base service which will be a Mongo DB service - a document-oriented database.

2.1.5 Communication interfaces

This system functions in a LAN of the bank environment. It does not essentially need Wide Area Network (WAN) to function. Since the whole system and the machine learning algorithms are deployed in the local machine, the whole process is done by a single system unit. The communication part is needed after a *high* probability of fraud activity is detected. The user may have setup automatic calls to respective personnel to act against the fraudster, or the user may manually enter personnel to be contacted. Anyways, this scenario needs a proper, reliable communication system. If the LAN fails, a crucial part of the module will fail, hence failing the whole system.

2.1.6 Memory constraints

External memory: Most of the model's functionality will depend on the training data. Training data will be only needed in the training phase. So, in the production environment there will be no huge external memory constraints. Since we would have a web application interface, there will be no need for setting up applications too.

Internal memory: Machine learning model will require a notable amount of internal memory since the analysis process needs memory power. A minimum 4GB RAM will be sufficient for the prediction and presentation purposes.

2.1.7 Operations

- User will need a login system to prove his identity to the system.
- User (admin personnel) will need a web portal (dashboard) to manage the machine learning modules.
- When the use clicks on the transaction-based fraud detection module icon, the user could see the currently ongoing operation on each and every POS (Point Of Service), ATM (Automatic Teller machine).

2.1.8 Site adaptation requirements

Since this application deals with sensitive client information, there should be high security measure taken. Other than that, there are no site-specific requirements for this application.

2.2 Product functions

Login and security

This application's data is confidential. If it gets leaked, there may be unnecessary problems. Due to this, we have to ensure a good security protocol to be followed throughout this system which will only allow the desired personnel to look at the data collected and the information passed.

Transaction module dashboard / portal

The dashboard will show some information that can be used to identify what is going on quickly. Information about the current transaction being analyzed, it's details, percentage of analyze, percentage of being a fraud up to now will be some quick tips. Below the

current transaction display, there will be some overall information with graphs and charts which will depict the functionality of the module in the past week / month / year.

Underlying machine learning algorithm

This is the heart of this module. Prediction of the fraudulent transactions is solely done by it. Required data will be gathered from the counters and the teller machines and they will be fed into the algorithm. ML model will predict the fraudulent behaviors of the transaction while learning new patterns and analyzing already learned patterns.

2.3 User characteristics

The typical user will be a bank customer who will be doing a transaction. Basically, this can be any type of adult person since almost all the people use bank account these days. Educated, professional or poorly educated people will be using this system module. But we do not need to make the user interface easily understandable and user friendly since our system will not be directly accessible by these users. Since it is an underlying system of the main transaction system of the bank, typical user will not even notice this system.

Even though the transaction system is used by a typical user, the dashboard of the underlying system will be used be specific bank personnel. They will be normal employees an elevated literacy level.

2.4 Constraints

Developers cannot access online API's since this application does not essentially have internet access. Therefore, online resources will be restricted.

2.5 Assumptions and dependencies

This application will be a web based one. That will remove most of the dependencies found in a normal application since it is separated from the operating system, and other services.

2.6 Apportioning of requirements

Machine learning model should be the highest priority task. It should be implemented using TensorFlow. Then the Data should be fed to the model. After that the model will be trained. User interface is the least prioritized task. It will be needed for the bank personnel.

3 Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

Login Page

Pre-determined personnel credentials are inserted in to the database beforehand. If the user is able to provide the given valid username and password, access will be granted to enter the system.

Transaction Analyzing Dashboard

After logging into the system, user can see some of the statistics of the transaction-based fraud detection system. Most importantly, the currently being analyzed transaction details will be shown in the top of the page. Below that, user can see some statistics of the predicted activities. Frauds detected, non-frauds detected counts will be shown and can be sorted according to user's desire.

Alert Popup

This is the most important part of the module. Alerts of this kind will be shown in the pre-selected personnel devices. The transaction details can be seen via the alert box by clicking more details button. The Contacts button will navigate the user to a screen which can be used to contact personnel which he desires. All the contact details can be pre-configured by the developers.

3.1.2 Hardware interfaces

As stated above, this module does not have any specific hardware requirements. But it needs a functioning system unit at least with a core i5 central processor, 4gb ram for the predicting services of the model. Storage devices will not be a major requirement since the trained model is deployed in the client machine. (data sets are only needed at the training phase).

3.1.3 Software interfaces

TensorFlow framework will be used for training the machine learning module. After the training phase it will get the data stream from bank data and predict the probability of frauds.

For the admin personnel of the bank, we will be implementing a web application using Angular which will have a dashboard for the who system. All the modules of the system could be accessed by the Angular based web application.

Since both the frameworks (TensorFlow and Angular) are node package-based java script frameworks, it will be easier to communicate and handle within the software applications.

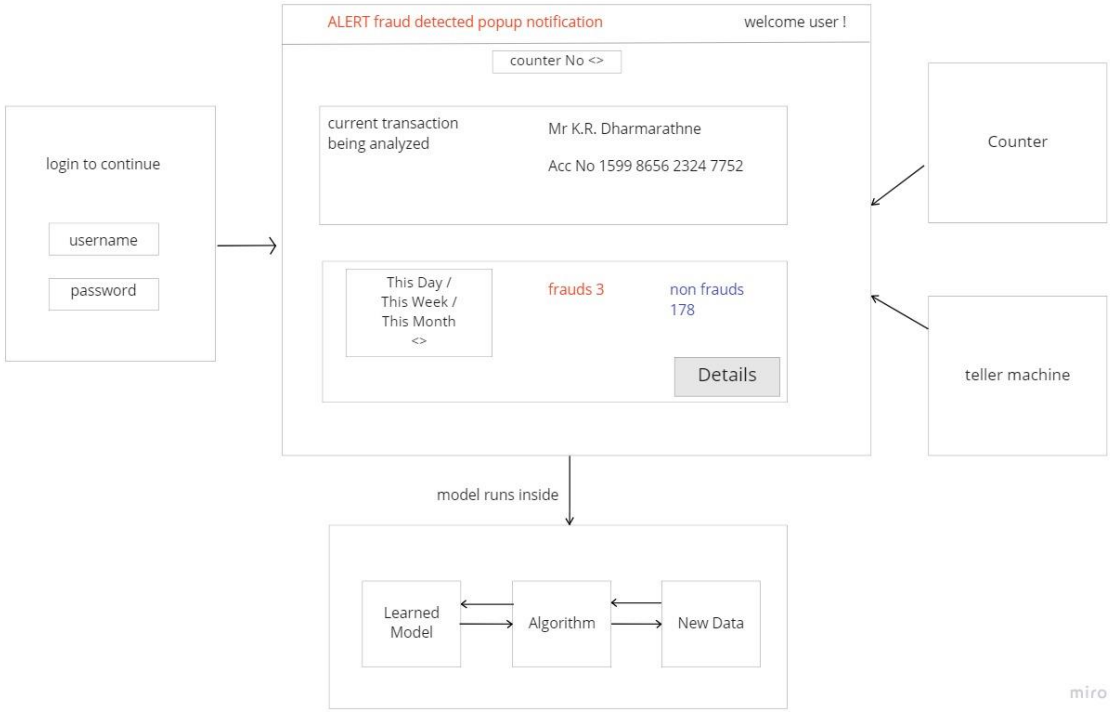
3.1.4 Communication interfaces

Wide Area Network (Internet) will not be a requirement for this system since we will be using the machine learning models for all the basic work. Because of that all the functionalities can be done in the local machines of the Bank premises. After the application is deployed to the local machine, only LAN (Local Area Network) connections will be needed for the system functionality.

After a possible fraud is detected, the application will need the ability to send alerts to required personnel. This function will need communication mediums. A locally connected network is sufficient for this.

3.2 Architectural Design

3.2.1 High level Architectural Design



3.2.2 Hardware and software requirements with justification

Basic hardware requirement is to have a functional local machine. It will need a medium core i5 processor and a 4GB ram. That is because after training the model, we will not need much processing power to predict the results. But to train the model, we might need powerful machines.

Angular, TensorFlow, MongoDB stack will be used for the software requirement. TensorFlow is a easy to use, popular machine learning framework which can be used in node packages. Since of that, this stack will be easy to use unlike other python based frameworks.

3.2.3 Risk Mitigation Plan with alternative solution identification

In any plan, there might be places where it can all go wrong. Some places we did not think of before. Even though the best type of planning was done, this happens inevitably. There are many other alternatives in case of these scenarios. Other technology stacks like React, Flask, SQL Server will be potential plans. Other UI arrangements could also be discussed to use instead of the dashboard method used in here. Setup of the hardware devices could also be adjusted according to the intention of developers.

3.2.4 Cost Benefit Analysis for the proposed solution

There will be costs for hardware devices which is needed to identify customers and their various expressions. But these hardware will be crucial for the research because they are a must for this research. So the cost of them is negligible compared to the benefit we get.

3.3 Performance requirements

Memory Allocation

A small disk space will be sufficient for the application since the machine learning model does not acquire a large space in the hard disk. Most of the data is needed for the training phase only. Not for the deployment phase. So, after training the model and deploying the application in the localhost of the bank, only a powerful processing unit and a RAM will be needed. Approximately, 1GB of space, 4Gb of RAM, core i5 processing unit will be sufficient to execute the application.

Response Time

After a transaction is initialized in a counter or a teller machine, the transaction analyzing begins. Since that moment, till the flag is set as fraud or not, time should be considered as a one scenario.

And the second scenario where the time should be measured is, alerting the required personnel. After a transaction has been flagged as a fraud, till all the registered personnel gets the alert, time should be measured. Since the bank personnel is connected through a LAN, this should be done at the bank premises.

Workload

The number of counters and the teller machines (all the point of services which have the ability to do transactions) plays the major role here. According to the number of transactions and the system's ability to keep up with them, a peak workload should be defined.

Scalability

The workload scalability might not vary frequently since this is not a publicly open service through internet. This service is only used at a given bank premises. But, when the bank customers get to increase, that should be taken into account. This information can be observed from the bank past data.

3.4 Design constraints

Using the industry best practices may make the experience more pleasant for the users. Navigation, UI designs is important to be looked at. Because the bank personnel will not be IT professionals, this machine learning model-based solution should be introduced as easy to use through a user-friendly web application. So, this application should be developed as if any person can use it irrelevant of their literacy levels.

3.5 Software system attributes

3.5.1 Reliability

This module should be predicting the fraud possibilities at least at a rate of 80%. If it becomes lower than that, the commitment of this module to the system will be very low. Because from all the four modules, most reliable modules will be giving a higher percentage to the decision of fraud detection. For a competitive product, this should be in a higher level of reliability. This could be measured by confusion matrices and other techniques, by predicting and later resolving whether the predictions were correct or wrong.

3.5.2 Availability

This system's availability mostly relies on the LAN performance of the bank premises. If the LAN is available, all the information streams are open and the system will function as usual. There are no dependencies outside the premises i.e. internet connection.

Since the local host will have the minimum performance skills required, there will be no delay of analyzing the transactions real time.

3.5.3 Security

Security is a major concern in this context of the application. Banking applications are the most vulnerable for attacks. But since we do not expose our services to the WAN, there is no vulnerabilities available to be penetrated. Only security concerns are the normal physical attacks. Other than that, LAN's can not be penetrated through outside parties, giving the application whole lot of security.

If an outside person gains access to the local host which our application is deployed, he still should bypass our login system to access the sensitive information. Other than that, there is no way to access bank users' personal sensitive information through our application.

3.5.4 Maintainability

The machine learning model is the only components that changes frequently and automatically. So, there is nothing programmers can do. The model will self-learn patterns while the transaction information is fed through real time day to day transactions.

3.6 Other requirements

There are no other specific requirements to be mentioned.

4 Supporting information

4.2 Appendices

4.3 References

[1] Deploying a machine learning model. Medium. [Online]. Available: https://medium.com/@dvelsner/deploying-a-simple-machine-learning-model-in-a-modern-web-application-flask-angular-docker-a657db075280

[2] Machine learning: TensorFlow with Angular. Medium. [Online]. Available: https://medium.com/@dhormale/integrate-machine-learning-tensorflow-model-with-angular-d72ec9287520

[3] Riskified – A fraud detection technology. [Online]. Available: https://www.riskified.com/resources/risk-academy/fraud-detection-technology/