# Request for Proposal

**REQUEST FOR PROPOSAL**

***Credit Card Fraud Detection System***

***2018-03-20***

***Michiel van der Merwe & Michael Combrinck***

***Client: Raymond H***

Contents

[Request for Proposal 1](#_Toc510375682)

[Executive Summary 3](#_Toc510375683)

[Proposed System: 4](#_Toc510375684)

[Problems Solved/Capabilities: 4](#_Toc510375685)

[Characteristics: 4](#_Toc510375686)

[Environment: 4](#_Toc510375687)

[Proposed Timeline 5](#_Toc510375688)

[Project Purpose and Description 5](#_Toc510375689)

[Project Scope 7](#_Toc510375690)

[Design: 7](#_Toc510375691)

[Functional Requirements: 7](#_Toc510375692)

[HMM for Credit Card Detection 8](#_Toc510375693)

[HMM Model for credit card transaction and Neural network 8](#_Toc510375694)

[Technical Requirements: 10](#_Toc510375695)

[Organizational Requirements: 12](#_Toc510375696)

[User Interface Design: 12](#_Toc510375697)

[Database Design: 13](#_Toc510375698)

[Plastic/Card Record Layout 13](#_Toc510375699)

[Account Record Layout 15](#_Toc510375700)

[Transaction Record Layout 16](#_Toc510375701)

[Customer Record Layout 17](#_Toc510375702)

[Development: 20](#_Toc510375703)

[Coding: 20](#_Toc510375704)

[Naming Conventions and Standards 20](#_Toc510375705)

[Good Programming Practices 24](#_Toc510375706)

[Licensing: 27](#_Toc510375707)

[Additional: 27](#_Toc510375708)

[Included: 27](#_Toc510375709)

[Not Included: 27](#_Toc510375710)

[Background of Proposed System 27](#_Toc510375711)

[Audience 27](#_Toc510375712)

[Tools and Functionalities 28](#_Toc510375713)

[Essential components 29](#_Toc510375714)

[Non-Essential components 30](#_Toc510375715)

[Reporting Needs 30](#_Toc510375716)

[Proposed System Specification 30](#_Toc510375717)

[10. Available technology resources/integration issuers 31](#_Toc510375718)

[11. Budget & Cost estimates 31](#_Toc510375719)

[12. Proposal Evaluation Criteria/ Format for proposals 32](#_Toc510375720)

[2. Technical Volume 32](#_Toc510375721)

[3. Management Volume 33](#_Toc510375722)

# Executive Summary

**What you offer and the problem your business solves -**  This product is very necessary because credit card fraud is still a very big problem this will provide a lot of business value seeing as the product development will not cost the organisation any money but the finished product can be sold for a lot of money

**Your target markets -**The suggested audience for this system is banking companies that have had problems with fraud before and were not able to detect it. The stakeholders of our project are our client Raymond H. who will be regularly checking the system and provide more needs. The entire system will be available to the general public, but it can only be used once paid for.

**Business plan purpose.** Say whether you’re seeking investment or trying to secure a bank loan. An executive summary is only really necessary when you are sharing your business plan with outsiders.

**Size or scale** – The current scale of the project will be small but the when the product is done the scale will become massive maybe even an industry standard.

# 

|  |  |
| --- | --- |
|  | Proposed System: The proposed system is going to a credit card fraud detection system. This system will scan user behavior when using their credit cards and will also use location scanning to check for unusual patterns. These patterns include:   * User spending patterns   If an unusual pattern is detected the system requires reverification. The system analyses user credit card data for the following characteristics:   * Country * Unusual Spending Procedures   The system will use historical data to determine an unusual spending pattern and may require the user to login again or ban the user after a number of invalid attempts. |
|  | Problems Solved/Capabilities: This system will provide a safer way of making transactions. It will minimize fraud which is an increasing problem in our country. It will provide a sense of security to multiple organizations that will use the system. Once created, the system can in the future, be able to integrate with advanced security such as biometrics. It can also provide a customer of a credit card information on whether their card has been cloned or is being used without them knowing, then the customer can inform the system that they want to block the card until the issue is resolved. Characteristics: The system will need to make use of the following:   * SQL or oracle type Relational Database * Python Programming Language for the neural network * The output will be provided using a website * The website will be designed using Angular |
|  | Environment: Hardware:   * Card machine * Computers the developers will be using for development and testing as well as presentation   Software:   * Operating System being windows 10 * Visual studio 2017 for creating a Python Neural network * Photoshop and anther photo editing tools * Data analysis tools like Microsoft excel * SourceTree   Services:   * GitHub |
|  |  |

# 

# Proposed Timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Description:** | **Deadline:** | | Develop Request for Proposal | 2018-03-20 | | Request for Proposal Release Date | 2018-03-21 | | Submissions of Question on Request for Proposal | 2018-03-28 | | Notification on approval of Request for Proposal | 2018-03-29 | | Answers to Questions | 2018-03-31 | | Initial Meeting | ? | | Drop-dead-Date (No new concepts/functionality) | ? | | Client Selected | Completed | | Proposal Approved | ? | |  |

# Project Purpose and Description

The purpose of the project is as follows: Providing a way of reducing fraud in credit card use in the world.

Description: The system will use and user behavior to search for unusual activity when using a credit card. The system will analyze various characteristics such as country, unusual spending procedures and historical data. Based on historical data the system will be able to recognize unusual buying patterns and the system will need reverification. This can be done by asking the user to log in again, or the user can also be blocked after a number of failed attempts. Some of the functional requirements the system will need to include are:

* Database: Normalized and containing the history of the client, location and buying patterns.
* Mathematical Validation and Verification: Statistically analyzing the data
* Reverification Methods
* Log In: User and Admin
* CVV Code check on back of card.
* Verification SMS or email to determine if the transaction was mad by the user
* Security methods: So that the system cannot be overridden by a hacker’s code
* Business Intelligence

# 

# Project Scope

## Design:

## Functional Requirements:

Capability and reduce financial loss, Credit Card Fraud Detection with Artificial neural network System card watch: A Neural Network based database mining index terms—Electronic Commerce, Credit card fraud, Artificial Intelligence, Artificial Neural Networks, Sequence System for Credit Card Fraud Detection which is bases Alignment, Machine Learning.

To establish a smooth channel of data exchange, web services techniques such as SML, SOAP, and WSDL are used.

Create Clusters

Identify spending profile of the cardholder

Choose initial set of probabilities based on the spending profile

Construct sequences from training data

Construct and train model

Training

Detection

Anomaly

Normal

Generate observation symbol Og-1

Add Og-1 in existing sequence to form new sequence

Add OR-1 Symbol to the existing sequence

Test

Alarm

Accept both old and new sequence

Calculate ∆ᶞ

**Hidden Markov Model (HMM)**-based credit card FDS, is the one that we perceive as the most ideal because it does not require fraud signatures and yet is able to detect frauds by considering a cardholder’s spending habit. We model a credit card transaction processing sequence by the stochastic process of an HMM. The details of items purchased in individual transactions are usually not known to an FS running at the bank that issues credit cards to the cardholder. This can be represented as the underlying finite Markov chain, which is not observable. The transactions can only be observed through the other stochastic process that produces the sequence of the amount of money spent in each transaction. Hence, we feel that HMM is an ideal choice for addressing this problem. Another important advantage of the HMM-based approach is a drastic reduction in the number of false positives transactions identified as malicious by an FDS although they are actually genuine.

## HMM for Credit Card Detection

An FDS runs at a credit card issuing bank. Each incoming transaction is submitted to the FDS for verification. FDS receives the card details and the value of purchase to verify whether the transaction is genuine or not. The types of goods that are bought in that transaction are not known to the FDS. It tries to find any anomaly in the transaction that is not known to the FDS. It tries to find any anomaly in the transaction based on the spending profile of the cardholder, shipping address, and billing address, etc. If the FDS confirms the transaction to be malicious, it raises an alarm, and the issuing bank declines the transaction. The concerned cardholder may then be contacted and alerted about the possibility that the card is compromised.

## HMM Model for credit card transaction and Neural network

To map the credit card transaction processing operation in terms of an HMM, we start by first deciding the observation symbols in our model. We quantize the purchase values x into M price ranges V1,V2 and VM, forming the observation symbols at the issuing bank. The actual price range for each symbol is configurable based on the spending habit of individual cardholders. These price ranges can be determined dynamically by applying a clustering algorithm on the values of each cardholder’s transactions.

The Artificial Neural Network (ANN) module is a VB standard module that takes care of the artificial neural network algorithm for classifying data based on their attributes. Given a dataset containing any number of data points, the algorithm separates the dataset into groups of similar data points. It takes as input the number of groups or clusters to separate a dataset into and a dataset containing the items to be clustered or separated into groups. Each data point in the dataset is assigned to the cluster and stored in the database.

The card fraud alert is designed in a way that alerts are generated when the selected risk exceeds a threshold. By adjusting the threshold, the number of alerts was controlled. As the threshold is lowered, the number of alerts increases. Consideration is made so as to remove any resulting deviation from accurate and reliable result because at the extreme, if the threshold is reduced to zero then every transaction will be alerted and although a 100% detection rate could be claimed there will also be an overwhelming number of false positives. The designed system supports several different types of alert and each has different priority levels of the four clusters used but two of the four clusters are adjusted to one.

The levels used are low, medium and high-risk. By setting the thresholds for medium level alerts to the optimal value, it can then set thresholds for the high-risk level alerts to catch high-risk incidents and the low-level alerts to seep up the low-risk ones. By differentiating alerts levels in this way, users can target resources and opt to be informed by email or SMS of particular alert types and levels. The available alerts introduced are system, pattern and customer alerts.

The risk engine generates system alerts and the computation of the risk measures was based on the systems built in algorithms, data. SQL is used to create and manage the database. Each of the data tables, Legitimate and fraudulent tables, are stored in a separate file. In the cluster name table, four cluster names were used. Since the binary classifiers used in most fraud detection software sometimes take legitimate transactions as fraudulent, the decisions to use four cluster names are to eliminate the problem. The cluster names are low, high, risky and high risk. The cluster tables to be used can be from four to any number but not more than ten.

After each transaction, the data point entry and clusters made are processed by the SOMNN engine and sent into the database. This helps the detection engine to known when any data entry is legitimate or fraudulent, and the reason is given immediately after the alert. Artificial neural networks are massively parallel-distributed processor that has the natural propensity for storing experiential knowledge and making it available for use. The processes of ANN comprise three stages such as training, testing, and deployment. There are two types of NN training methods- supervised and unsupervised methods. Neural networks are an extension of risk scoring techniques and are based on the statistical knowledge contained in extensive databases of historical transactions, and fraudulent ones in particular. These neural network models are basically trained by using examples of both legitimate and fraudulent transactions and are able to correlate and weigh various fraud indicators to the occurrence of fraud.

This network contains two layers of nodes – an input layer and a mapping layer in the shape of a two-dimensional grid. SOMNN component learning is a learning procedure that divides a set of input patterns into clusters that are inherent to the input data. The concept of fraud detection has been founded on data mining techniques such as association rules and classification. Research on fraud detection has been focused on pattern matching in which abnormal patterns are identified from the normally. To map the credit card transaction processing operation in terms of an HMM, we start by first deciding the observation symbols in our model. We quantize the purchase values x into M price ranges V1,V2 and VM, forming the observation symbols at the issuing bank. The actual price range for each symbol is configurable based on the spending habit of individual cardholders.

These price ranges can be determined dynamically by applying a clustering algorithm on the values of each cardholder’s transactions. The artificial neural network (ANN) module is a VB standard module that takes care of the artificial neural network algorithm for classifying data based on their attributes. Given a dataset containing any number of data points, the algorithm separates the dataset into groups of similar data points. It takes as input the number of groups or clusters to separate a dataset into and a dataset containing the items to be clustered or separated into groups. Each data point in the dataset is assigned to the cluster and stored in the database.

The data for credit card fraud from the database. In the output database interface, when a transaction data is checked, the software provides two ways of presenting detection results namely. Detection with all details provides step-by-step visualization of the checked transactions such as the cluster it belongs, the tables either suspect or legitimate. Detection without details option does not show certain details, e.g. clusters. It merely presents the result of the detection. The database was meant to run at the background of the existing banking software and be getting its data from real-time banking transaction, checking whether the transaction is legitimate or fraudulent.

## Technical Requirements:

This document is written to clarify the technical part of our project which includes the following topics…

* The project architecture.
* The considerations made with the design of the project in mind.
* The system characteristics of the project.
* The design method/s and standard/s used throughout the project.

Thus, to give a small description of our project…

Our system is going to feature user behavior and location scanning to check for unusual patterns. These patterns will include user characteristics such as user spending patterns as well as usual geographic locations to verify his/her identity. If any unusual pattern is detected, the system will require reverification. The system analyses user credit card data for various characteristics. These characteristics will include user country and also the user’s usual spending procedures. Based upon previous data of that user the system recognizes unusual patterns in the payment procedure. So now the system may require the user to login again or even block the user for more than 3 invalid attempts.

* Python Programming language

The reasoning behind the use of this language is because of the fact that most of the project team has more than one-year experience with using this language and because of the fact that this language is versatile enough for the purpose of our project.

* AngularJS

Because we want our project to be cross-platform we surmised that the best and most effective way to do this would be to build a web application, thus the main reason for using this language is because of the fact that it is really versatile when it comes to web design and building web applications.

When it comes to the design of our project we had some considerations to make in regards with the look and feel of the project as well as when it came to the way our team communicates during the project…

* Team communication

When it came to deciding how the team would communicate during the project we had to decide between physical meetings or electronic ones by using VOIP applications/Video call applications, we also needed to decide on the frequency of our meetings. In the end we decided that using the “Discord” VOIP application for our meetings would be best especially because everyone lives so far from one another, we also decided that we would have a meeting once a week to keep the project team focused and on track.

* Project look and feel

When it came to the project look and feel we had to decide on the language that we wanted to use when it came to the design of our project and we also had to decide whether we wanted to decide on the design of the project together or if we wanted one person in charge who could delicate tasks to make his vision for the design of the project a reality. In the end we decided on the AngularJS language for the design of the project because of the fact that it is so versatile and because it works so well when it comes to creating web applications, we also decided that placing one person in charge of that task would benefit us more than deciding on the design together because of the fact that our ideas would clash it we wouldn’t reach a conclusion.

## Organizational Requirements:

These requirements are the requirements that the organization is responsible for to ensure the wellbeing of its employees.

Requirements for employees:

* Time: It is expected that all employees arrive on time for meetings and that the assigned work is completed on the due date specified.
* They should get along with others.

The organization has the following responsibilities:

* Educate the workforce so that it understands the organization's mission and the organization's clients.
* Match capabilities of workers to ensure they enjoy doing the job.
* Train workers.
* Provide necessary instructions so team knows what needs to be done and how it needs to be done.
* Provide workers with equipment to do the job.
* Evaluate performance to find weaknesses and correct them.

Note: Safety will not be considered as some organizational requirements as we are using Git Hub to do the project from our own homes, so physical security is not needed.

The system should also be able to adhere to many organizations requirements such as banks so that the whole structure that they are working on does not need to be changed.

## User Interface Design:

This design is strictly preliterary



## Database Design:

### Plastic/Card Record Layout

The following table is the common structure of a Card in the bank. The green fields are NOT on the record on a bank’s database but stored more securely. Passwords, CVV and Expiry date are never stored in a database, but rather hardware stored. For the purpose of this project, I would recommend storing that data on the database under the card record to keep it simple.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Field** | **Size** | **Description** |
| PK | Plastic Number | 23 | The plastic is generally a 16 character field with the rest being leading zeroes for example 00000004512368512264512.   The first 3 numbers are registered to the bank by either Visa or Mastercard. In the example above it will be 451.  The next 3 identifies the product, so those 3 numbers must correspond with the product field above. In the example above it will be 236.  As you can see, the last 3 digits (512) of the plastic number identifies the card in the account range.  The last digit is a checksum digit. |
| FK | Company ID | 03 | Generally used to distinguish which division in a bank the card belongs to. An example is having a branch in RSA and one in Bot. Both countries will have different codes. We can make up our own. If we only use RSA, we do not need this field. |
| FK | Product | 03 | Used for main product classifications like General (GEN), Corporate (CRP) and Business (BUS) |
| FK | Sub-Product | 03 | Generally used to indicate which product house this card belongs to PLA for a platinum card for example or GLD for gold card. |
| FK | Account Number | 23 | Account number linked to this plastic generally it will look like this: 00000004512368512264000. The last 3 digits are generally zeroes. This is to make space for cards. |
|  | Account Type | 01 | Primary Card (P) or Secondary (S). An account must have a primary before a secondary card can be opened. |
|  | Verified by Visa Indicator | 01 | Active (A), Inactive (I). This is used for if a card has been activated for Visa verification on online purchases. |
|  | Primary Plastic Number | 23 | The primary plastic number which the secondary card is attached to. This number must only be populated if this record is a secondary plastic. |
|  | Current Status | 03 | This is the status of the card at this moment. If it is inactive, open or deleted. |
|  | Previous Status | 03 | This is what the current status was before it is changed. This is to make sure that the card statuses do not get messed up. For example, you cannot have an expired/delete previous status, but the current status is open. |
|  | Last Date Status Change | 08 | The last time the status of the card was changed |
|  | Emboss Line 1 | 40 | Emboss line 1 is generally empty and only used for business cards to add the company name to the card. |
|  | Emboss Line 2 | 40 | Emboss line 2 will always hold the name of the person to whom the card has been issued to. |
|  | CVV Number | 03 | This is the code used for online purchases. |
|  | Last Withdrawal Limit Change | 08 | The last time the credit limit was changed |
|  | Card Created | 08 | The date the card was created |
|  | Card Opened |  | The date the card was activated/have been opened. |
|  | Card Expiry date | 06 | When will the card expire (format mm/yyyy). On the card the format Is only mm/yy |
|  | Password | 20 | The password of this plastic. |
|  | Reissued Card Number | 23 | If this card is replaced, this field must be populated with the next card. Otherwise this field is empty. |
|  | Reason | 40 | If plastic was closed/cancelled or deleted, this field must contain the reason for the action. |

### Account Record Layout

This record layout is the layout of an account. There is a lot of data, but you can decide what we need and what we can remove. If you do not understand a field, feel free to ask.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Field** | **Size** | **Description** |
| PK | Account Number | 23 | The plastic is generally a 16 character field with the rest being leading zeroes for example 00000004512368512264512.   The first 3 numbers are registered to the bank by either Visa or Mastercard. In the example above it will be 451.  The next 3 identifies the product, so those 3 numbers must correspond with the product field above. In the example above it will be 236.  The last digit is a checksum digit. |
| FK | Company ID | 03 | Generally used to distinguish which division in a bank the card belongs to. An example is having a branch in RSA and one in Bot. Both countries will have different codes. We can make up our own. If we only use RSA, we do not need this field. |
| FK | Account Product | 03 | Used for main product classifications like General (GEN), Corporate (CRP) and Business (BUS) |
|  | Account Type | 04 | Cheque (Cheq), Savings (Save), Transmission (Tran), Bond, Subscription Share (Sub) |
|  | Branch Code | 06 | Branch code of the branch where the account was opened |
|  | Inactive Cards | 03 | Number of inactive cards on account. |
|  | Open Cards | 03 | Number of open/active cards on the account |
|  | Closed Cards | 03 | Number of closed/canceled/destroyed cards on this account. |
|  | Primary Plastic Number | 23 | The primary plastic number associated to this account. |
|  | Current Status | 03 | This is the status of the card at this moment. If it is inactive, open or deleted. |
|  | Previous Status | 03 | This is what the current status was before it is changed. This is to make sure that the card statuses do not get messed up. For example, you cannot have an expired/delete previous status, but the current status is open. |
|  | Last Update Date | 08 | Last date the account was updated |
|  | Last Update Time | 06 | Last time the account was updated |
|  | Last Update Operator | 08 | The last Operator ID that updated the account. This can be null if the terminal is online. |
|  | Last Update Terminal | 08 | The last terminal that updated the account. This can be an online update or an update done in a bank branch |
|  | Account Open Date | 08 | The date the account was opened |
|  | Account Close Date | 08 | The date the account was closed |
|  | Cycle Date | 08 | The cycle date of the account. When statements must be done and fees must be paid. |
|  | Account Expiry Date | 08 | The date the account will expire. |
|  | Initial Issuing Date | 08 | The date which the first credit card was issued on this account. |
|  | First Activity Date | 08 | The date which the first transaction was done. |
|  | Last Known Activity Date | 08 | The last date any action was done on this account. |
|  | Last Reissuing Date | 08 | The last date a card was reissued on this account. |
|  | Last Fraud Activity Date | 08 | The last date fraud was detected on the account |
|  | Last ATM Activity Date | 08 | The last time an ATM was used for this account |
|  | Last Auth Required Date | 08 | The last date an authorization was required on this account. |
|  | Last Auth Approval Date | 08 | The last date an authorization was approved on this account. |
|  | Outstanding Auth Amount | 11.2 | The total amount of auths that are outstanding |
|  | Outstanding Auth Count | 07 | The total number of auths that are outtstainding |
|  | Outstanding Extended Credit Auths | 07 | The total number of auths outstanding on budget transations |
|  | Auths Approved | 03 | The total number of auths approved |
|  | Auths Declined | 03 | The total number of auths declined |
|  | Last Batch Process Date | 08 | The last date this account was processed via batch |
|  | Last Transaction Date | 08 | The last date there was a transaction done on this account |
|  | Last Transaction Amount | 07 | The last transaction amount done on this account |
|  | Total Transactions | 03 | Total transactions done over this account |
|  | Reason | 40 | If account was closed, this field must contain the reason for the action. |

### Transaction Record Layout

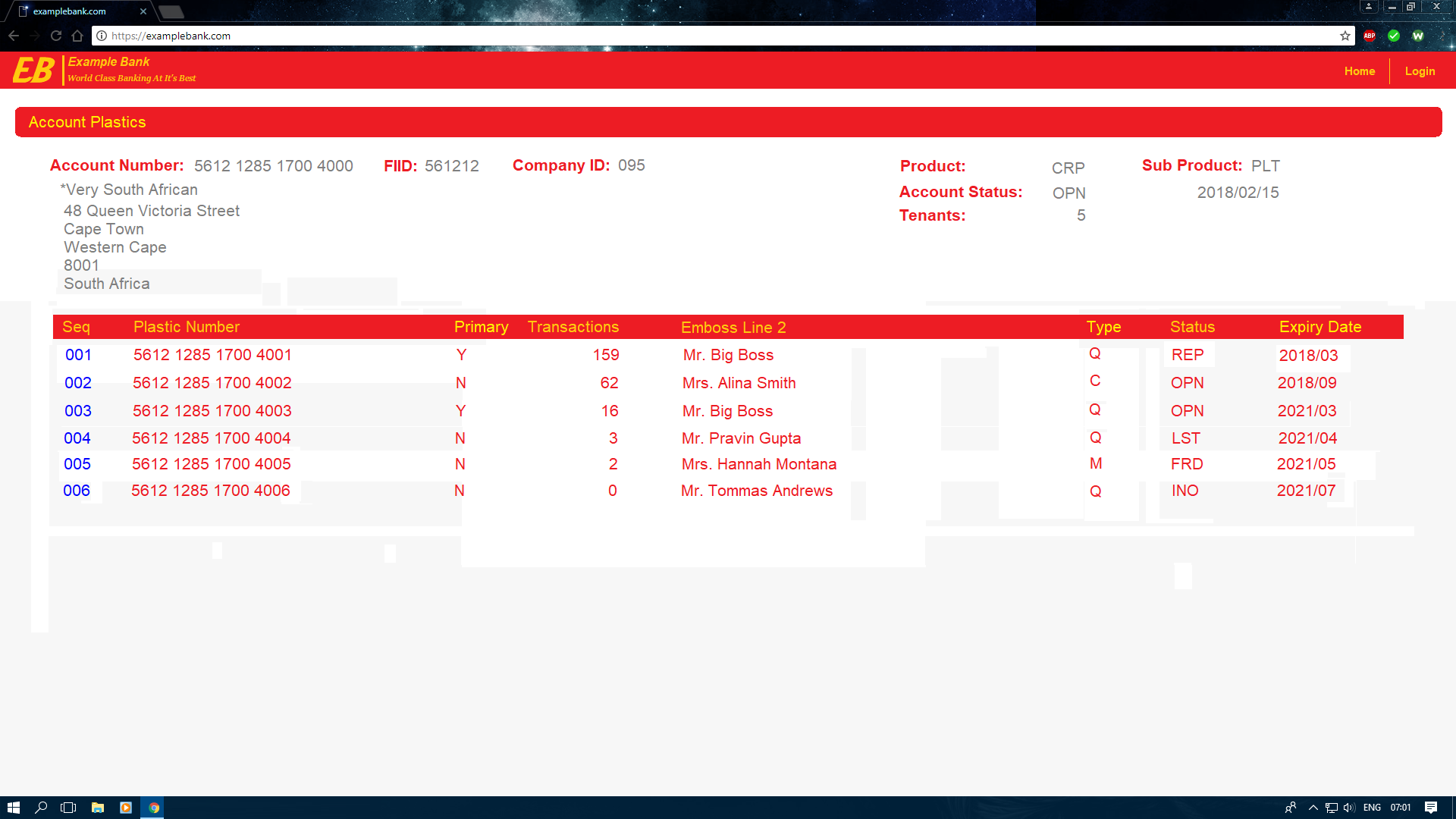
This is the record that holds the transaction data for a plastic.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Field** | **Size** | **Description** |
| PK | Auth Code | 05 | The Auth code that gets passed by a Point of Sale Device it is used in the place of the Transaction Code if there are no transaction code. |
| PK | Transaction Code | 05 | The transaction code that gets passed by a Point of Sale Device |
| FK | Company ID | 03 | Generally used to distinguish which division in a bank the card belongs to. An example is having a branch in RSA and one in Bot. Both countries will have different codes. We can make up our own. If we only use RSA, we do not need this field. |
| FK | Account Product Code | 03 | Product code of the account. Used as part of the key. |
| FK | Account Number | 23 | Account number on which the transaction was done on. |
| FK | Plastic Number | 23 | Plastic number on which the transaction was done on. |
|  | Transaction Status Indicator | 01 | Typical Status indicators = Information Only (I), Normal Item (N), Disputed Item (D) |
|  | Transaction Source Indicator | 01 | MasterCard (M), Visa (V), Local Bank (L), Competitor Local Bank (C) |
|  | Transaction Source | 05 | Merchant Code |
|  | Source Currency Code | 03 | The currency in which the transaction was done from |
|  | Destination Currency Code | 03 | The currency in which the transaction must process. Generally in our case it is in Rand (ZAR) |
|  | Transaction Date | 08 | Date on which the transaction was done. |
|  | Transaction Time | 06 | Time the transaction was done. |
|  | Original Posting Date | 08 | The date which the transaction was posed (processed) from an Auth |
|  | Transaction Type | 03 | Payment/Settlement (PU), Auth (AU), Auth Reversal (AUR) |
|  | Transaction Amount | 11.2 | The amount of the transaction |
|  | Adjustment Reason Code | 03 | The code for making an adjustment to the amount. Can only be done from the Transaction source. |

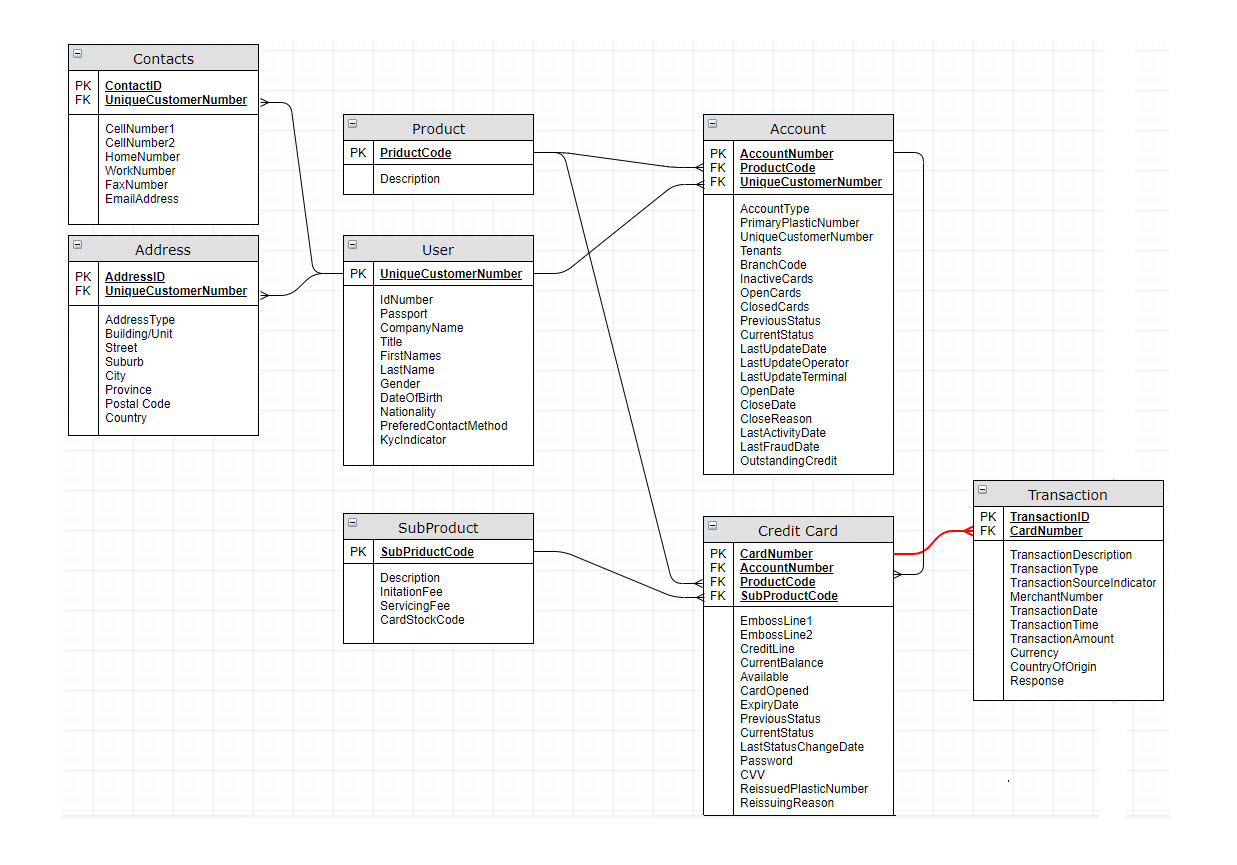
### Customer Record Layout

The user information is not in my field of work. Our Customer Information Systems division holds this data. I will try my best to provide you with the most detailed information on the customer information required.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Field** | **Size** | **Description** |
| PK | Unique Customer Number | 10 | This is a unique number generated for every customer |
|  | Title | 04 | Title of the customer |
|  | Name | 40 | Name of the customer |
|  | Surname | 40 | Surname of the customer |
|  | Gender | 01 | Male (M), Female (F) |
|  | Nationality | 01 | Foreign (F), Local (L) |
|  | Date of Birth | 08 | Customer’s date of birth |
|  | ID | 13 | This is the id number of the customer. It is only applicable if the customer is a South African national |
|  | Passport Number | 15 | This is the passport of the customer. It is only applicable if the customer is a foreigner. |
|  | Cellphone Number | 13 | The cellphone number of the customer. |
|  | Home Number | 13 | The home number of the customer. |
|  | Work Number | 13 | The work number of the customer. |
|  | Fax Number | 13 | The fax number of the customer. |
|  | Email Address | 25 | Email address of the customer. |
|  | Preferred Contact Method | 01 | Email (E), Cellphone (C), Home phone (H), Work phone (W), Fax (F), Post (P) |
|  | Company or Business Name | 40 | Only applicable if this is a company or business user. |
|  | Address Line 1 | 40 | Address Line 1 |
|  | Address Line 2 | 40 | Address Line 2 |
|  | Address Suburb | 40 | Suburb |
|  | Address City | 25 | City |
|  | Address Province | 40 | Province |
|  | Address Postal Code | 04 | Postal Code |
|  | KYC Indicator | 01 | Verified (V), Unverified (U), Expired (E). KYC stands for Know your Customer. It is the documents which the bank asks for, like the Proof of Residence and the Certified ID for example. This is set manually in the branch. |







## Development:

The system is going to be developed by a third-year group at Belgium campus with project manager Michiel van der Merwe. The developers will make use of their own personal computers to create the systems. Weekly meeting will be held on either Monday or Thursday where the technical walkthrough of the week will be discussed. We will follow the prototype model to design and develop the project. This process includes the product being built, tested and refined if any errors occur. It is seen as an iterative, incremental, trial-and-error process.



## Coding:

The coding language that will be used is Python as most developers are familiar with it. Coding best practices will be enforced such as:

The software will be written in Python and certain application program interface or ApI’s for short will be used, although the code snippets below are aim towards C# language it is based on object-oriented programing so one still needs to follow the guide line, the developer’s and project manager must make sure that Object oriented programming standards are followed this includes:

### Naming Conventions and Standards

Pascal casing: the first character of all words is upper case and the other characters are lower case.

Camel casing: the first character of all words, except the first word, is upper case and other characters are lower case.

Use Pascal casing for class names:

Hide   Copy Code

public class HelloWorld

{

...

}

Use Pascal casing for method names:

Hide   Copy Code

public class HelloWorld

{

void SayHello(string name)

{

...

}

}

Use Camel casing for variables and method parameters:

Hide   Copy Code

public class HelloWorld

{

int totalCount = 0;

void SayHello(string name)

{

string fullMessage = "Hello " + name;

...

}

}

Do not use Hungarian notation to name variables. In earlier days, most programmers liked it: having the data type as a prefix for the variable name and using m\_ as the prefix for member variables, e.g.:

Hide   Copy Code

string m\_sName;

int nAge;

However, in .NET coding standards, this is not recommended. Usage of data type and M\_ to represent member variables should not be done. All variables should use Camel casing. Use meaningful, descriptive words to name variables:

* Do not use abbreviations. Use name, address, salary etc. instead of nam, addr, sal.
* Do not use single character variable names like a, n, x, etc. Use names like index and temp.

One exception in this case would be variables used for iterations in loops:

Hide   Copy Code

for ( int i = 0; i < count; i++ )

{

...

}

If the variable is used only as a counter for iteration and is not used anywhere else in the loop, many people still like to use a single char variable (i) instead of inventing a different suitable name.

* Do not use underscores (\_) in variable names.
* Namespace names should follow the standard pattern.

Hide   Copy Code

<company name>. <product name>. <top level module>. <bottom level module>

File name should match with class name. For example, for the class HelloWorld, the file name should be helloworld.cs (or helloworld.vb).

Indentation and spacing: use TAB for indentation. Do not use spaces.

Comments should be in the same level as the code. Curly braces ( {} ) should be in the same level as the code outside the braces. Use one blank line to separate logical groups of code.

Hide   Copy Code

bool SayHello (string name)

{

string fullMessage = "Hello " + name;

DateTime currentTime = DateTime.Now;

string message = fullMessage + ", the time is : " +

currentTime.ToShortTimeString();

MessageBox.Show ( message );

if ( ... )

{

*// Do something*

*// ...*

return false;

}

return true;

}

This code looks better than the code shown above:

Hide   Copy Code

bool SayHello ( string name )

{

string fullMessage = "Hello " + name;

DateTime currentTime = DateTime.Now;

string message = fullMessage + ", the time is : " +

currentTime.ToShortTimeString();

MessageBox.Show ( message );

if ( ... )

{

*// Do something*

*// ...*

return false;

}

return true;

}

There should be one and only one single blank line between each method inside the class. The curly braces should be on a separate line and not in the same line as if, for, etc.

Good:

Hide   Copy Code

if ( ... )

{

*// Do something*

}

Not good:

Hide   Copy Code

if ( ... ) {

*// Do something*

}

Use a single space before and after each operator and brackets.

Good:

Hide   Copy Code

if ( showResult == true )

{

for ( int i = 0; i < 10; i++ )

{

*//*

}

}

Not good:

Hide   Copy Code

if(showResult==true)

{

for(int i= 0;i<10;i++)

{

*//*

}

}

### Good Programming Practices

Avoid having too-large files. If a file has more than 300-400 lines of code, you must consider refactoring the code into helper classes. Avoid writing very long methods. A method should typically have 1-25 lines of code. If a method has more than 25 lines of code, you must consider refactoring it into separate methods. The method's name should tell what it does. Do not use misleading names. If the method name is obvious, there is no need of documentation explaining what the method does.

Good:

Hide   Copy Code

void SavePhoneNumber ( string phoneNumber )

{

*// Save the phone number.*

}

Not good:

Hide   Copy Code

*// This method will save the phone number.*

void SaveData ( string phoneNumber )

{

*// Save the phone number.*

}

A method should do only "one job." Do not combine more than one job in a single method, even if those jobs are very small.

Good:

Hide   Copy Code

*// Save the address.*

SaveAddress ( address );

*// Send an email to the supervisor to inform that the address is updated.*

SendEmail ( address, email );

void SaveAddress ( string address )

{

*// Save the address.*

*// ...*

}

void SendEmail ( string address, string email )

{

*// Send an email to inform the supervisor that the address is changed.*

*// ...*

}

Not good:

Hide   Copy Code

*// Save address and send an email to the supervisor*

*// to inform that the address is updated.*

SaveAddress ( address, email );

void SaveAddress ( string address, string email )

{

*// Job 1.*

*// Save the address.*

*// ...*

*// Job 2.*

*// Send an email to inform the supervisor that the address is changed.*

*// ...*

}

Use the C# or VB.NET specific types, rather than the alias types defined in the System namespace.

Good:

Hide   Copy Code

int age;

string name;

object contactInfo;

Not good:

Hide   Copy Code

Int16 age;

String name;

Object contactInfo;

Do not hardcode numbers. Use constants instead. Do not hardcode strings. Use resource files. Avoid using many member variables. Declare local variables and pass them to methods instead of sharing a member variable between methods. If you share a member variable between methods, it will be difficult to track which method changed the value and when. Use enum wherever required. Do not use numbers or strings to indicate discrete values.

Good:

Hide   Copy Code

enum MailType

{

Html,

PlainText,

Attachment

}

void SendMail (string message, MailType mailType)

{

switch ( mailType )

{

case MailType.Html:

*// Do something*

break;

case MailType.PlainText:

*// Do something*

break;

case MailType.Attachment:

*// Do something*

break;

default:

*// Do something*

break;

}

}

Not good:

Hide   Copy Code

void SendMail (string message, string mailType)

{

switch ( mailType )

{

case "Html":

*// Do something*

break;

case "PlainText":

*// Do something*

break;

case "Attachment":

*// Do something*

break;

default:

*// Do something*

break;

}

}

Do not make the member variables public or protected. Keep them private and expose public/protected properties. Never hardcode a path or drive name in code. Get the application path programmatically and use relative path. Never assume that your code will run from drive C: You never know; some users may run it from a network or from a Z: drive.

## Licensing:

The only licensing that can occur is if the campus decides they want to make the code totally private they will need to pay a GitHub private repository subscription of R90 a moth

## Additional:

* Research need to be done on security systems.
* Training will be providing to students with less knowledge of the coding language

## Included:

System that can perform the following:

* Verification
* Validation
* Unusual Patterns Tracking

## Not Included:

* Biometric Security
* Camera to Capture Card
* Physical System Built by Engineers

Background of Proposed System

The project manager Michiel van der merwe came up with an innovative project to help some form of struggle in the world. The proposal of a credit card fraud system was created, project manager was chosen and a team was formed on 2018 March. The team consists of 6 members including 2 Business Intelligence students and 4 Software Engineering students. The project Team includes:

* Michiel van der Merwe (Project Manager/BI)
* Zimkhita Komani (SE)
* Frederik Johannes Krige (SE)
* Arnoldus Snyman (SE)
* Michael Combrinck (BI)
* Akani Ngobeni (SE)

The team has decided in order to fulfill the business needs a neural network will be created in python programing language and for user interface purposes a website will be created using angular to display the results of the neural network

Audience

The suggested audience for this system is banking companies that have had problems with fraud before and were not able to detect it. The stakeholders of our project are our client Raymond H. who will be regularly checking the system and provide more needs. The entire system will be available to the general public, but it can only be used once paid for. There will be logins required in the system and the database including:

* Member
* Admin

When the member logs in they can access basic functionality such as checking previous transactions or payments but when an admin logs in they can check the purchasing history and geographical location and will have access to methods in the system that may not be accessed by the member.

Tools and Functionalities

|  |
| --- |
| Essential components  1. **Problem Statement** The need for the project is clear the way that the banks detect fraud is base on if this condonation is true then do this so what your group is going to design is a neural network that we will learn and then it is basically an autonomies system that will detect fraud. 2. **Methodology**. The type of design mythology that we will be using in our development will be the prototyping model because this is a project with a tight time frame this will be the best model to ensure that the system is done in time and working. 3. **Plan of Operation**.   First step will be to gather the requirements so that we stay with in the scope of the project  Then we will do a quick design to ensure that we are on the right track  Then the team will get feedback from the supervisors and the client to ensure that we are on the right track  Then the design will be refined to make sure that the final product is working properly.  Then finally the engineered product will be released and ready to use   1. **Time scales**, start date will be 16 April, milestone one which will be the front end and database will be done on 16 May, milestone to which is the neural net will be done on the 16 July then an extra 2 weeks to teach the neural net, website back end should be done by 16 August and will integration should be done by the end of August 2. **Project management:** The team has 2 major ways in which the members will be comminuting the first method being a WhatsApp group the second being a discord server (<https://discord.gg/BkB7PPU>) Both these methods of communications are free. third year students will come up with an innovative project to help some form of struggle in the world. The proposal of a credit card fraud system was created, project manager was chosen and a team was formed on March 2018, the project manager is Michiel van der merwe, Supervisors include Ms. Charmaine and Ms. Anila 3. **Risks,** there is no risk to the organization regarding this project only the project members the risk being that additional or no foreseen requirements gets added and the time runs out in terms of the schedule |
| Non-Essential components  1. **Costs,** there is some cost if the organization decides that they do not want to make the code public then they will need to pay a GitHub a private repository fee of R90/pm 2. **Business case,** the cost being so little and the benefit being that the organization can sell the system for a lot more that the cost meaning the potential business benefit is huge. 3. **Resources,** all the need resources and skills are available on campus for example if the team needs help with the math’s for the neural network our supervisor Ms. Charmain is a math lecture and can help us as well there are plenty of author math lectures as well that can help and if we are struggling with actually code MS Anila our author supervisor can help us. 4. **Impact**. The impact on the short term is very low but as the amount of data increase a need for a more reliable system will arise this is where our system comes into play the more data a neural network gets the more it learns this means that a increase in data does not mean that the system will struggle it will actually improve the system |
|  |
|  |

Reporting Needs

|  |
| --- |
| * User information such as name, surname * User spending amounts * Card information such as card numbers * Geographical information for the specific user’s card * Login details for users |

# Proposed System Specification

|  |
| --- |
| 1. Design parameters   These include the following…   * The actual app size must not be bigger than a 1 GB. * The app should be able to form a secure connection with the database used. * The app should be compatible with any Windows operating system starting from version 7. * The app should be designed in a way that is astatically pleasing and functional to the user. * The app should clearly state what input is asked of the user. |
| 1. Accessibility/Usability 2. Usability testing will validate if the app retrieves the correct information from the database and it will test the speed at which the app retrieves it. 3. Accessibility testing will test the UI part of the app and confirm if it is overcomplicated or actually usable. |

# 10. Available technology resources/integration issuers

|  |
| --- |
| 1. Genetic algorithms and other algorithms. Algorithms are often recommended as predictive methods as a means of detecting fraud. One algorithm that has been used in the past is based on genetic programming in order to establish logic rules capable of classifying credit card transactions into suspicious and non-suspicious classes. Basically, this method follows the scoring process. In the experiment described in their study, the database was made of 4,000 transactions with 62 fields. As for the similarity tree, training and testing samples were employed. Different types of rules were tested with the different fields. The best rule is the one with the highest predictability. Their method has proven results for real home insurance data and could be one efficient method against credit card fraud. |
| 1. Currently the banks (FMB) does not use a database to store any o the data of the card that was used. Thus, there is no database that will be needed to be imported so there will be no integration problems with the database. |
| 1. Yes, there is existing systems out there that do the same thing, all the banks use this system to ensure that credit card fraud is minimized but the team that will be work on this project will be using several new innovative techniques to ensure that the system will have a smaller error rate to ensure that customers satisfaction is increased |

# 11. Budget & Cost estimates

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Cost Item** | **QTY** | **Unit Cost** | **Total Cost** | **Maintenance first Years** | **Maintenance 5 years** | **Maintenance in %** |
| Private GitHub repository | 9 | R 90 | R 810 | None | None | None |
| **TOTAL COST** |  |  | R 810 |  |  |  |

# 12. Proposal Evaluation Criteria/ Format for proposals

|  |
| --- |
| **Expert assessment** – Reviewers should possess relevant knowledge and expertise in order to evaluate project proposals on the level of contributions to the broader scientific field which the project proposal belongs to, as well as in relation to specific objectives and methodology of a specific project proposal. Reviewers are selected based on clearly defined criteria.  **Transparency** – All decisions must be based on clearly described and publicly available rules, procedures and evaluation criteria. All project proposal applicants must receive adequate feedback the evaluation of their project proposals.  **Impartiality** – Project proposals are evaluated fairly and based on their quality. Conflicts of interest must be registered and processed according to established and public procedures. **Consistency** – evaluation procedure should be consistent with the Call type, adjusted to characteristics of scientific fields and in proportion with the value of project proposals and work complexity.  **Confidentiality** – All persons and organizations involved in the evaluation procedure must respect the confidentiality of all information listed in project proposals, including intellectual property, and all other documents. Integrity and ethical issues–  Ethics and integrity are the highest principles in the entire process of evaluation and their preservation is the responsibility of all persons involved in the evaluation |
|  |
| 2. Technical Volume  The system is going to be developed by a third-year group at Belgium campus with project manager Michiel van der Merwe. The developers will make use of their own personal computers to create the systems. Weekly meeting will be held on either Monday or Thursday where the technical walkthrough of the week will be discussed. We will follow the prototype model to design and develop the project. This process includes the product being built, tested and refined if any errors occur. It is seen as an iterative, incremental, trial-and-error process.   1. Standards and best practices have long been considered beneficial, as they stop us from designing based solely on personal opinion or experiences. To that end, the International Organization for Standardization (ISO) has created a set of common standards, called ISO 9241, which cover the Ergonomics of Human System Interaction. These standards have been developed by industry leaders through validated research and are intended to help designers use the best methods to generate findings we can rely on.   Several parts of .NET were made available under open source licenses, meaning any developer can contribute to it. For example, the Python compiler Roslyn was made open source by Microsoft under the Apache License. The source code of Roslyn can be downloaded from GitHub, and guidelines on how to participate and contribute were made available. Visual Studio is a very powerful Integrated Development Environment (IDE) where you actually write your Python or .NET programs. It’s popular because it supports code editing, interface design, server management, debugging, and performance analysis. You can download Visual Studio for free. .NET has a huge collection of predefined class libraries (pre-written code) that has support for simple and complex data structures. Essentially, that means you can rely on the work of hundreds of other developers and pull in already-written code into your own programs. .NET even has specific libraries for security, encryption, and database access. |
| 3. Management Volume  The team has 2 major ways in which the members will be comminuting the first method being a WhatsApp group the second being a discord server (<https://discord.gg/BkB7PPU>)  Both these methods of communications are free. third year students will come up with an innovative project to help some form of struggle in the world. The proposal of a credit card fraud system was created, project manager was chosen and a team was formed on 2018 March. The team consists of 6 members including 2 Business Intelligence students and 4 Software Engineering students. The project Team includes:   * Michiel van der Merwe (Project Manager/BI) * Zimkhita Komani (SE) * Frederik Johannes Krige (SE) * Arnoldus Snyman (SE) * Michael Combrinck (BI) * Akani Ngobeni (SE)   Special tools that will be use is visual studio for program development, discord VOIP services data analytics tools like Excel / Spreadsheet, Trifecta, Rapid Miner, Rattle, GUI, Qlikview, Weka, KNIME, Orange  Skill set that the team will need to complete the project will be a good project management, industry standard programming skills, good relational database practices like normalization.  Some sort of calculus background in order to do the calculations for the neural network.  This projects deadline is November 2018 the final product should be showcase in the mouth of November the milestones include complication of the front-end interface, middle tier of the program, business logic and the database and or log files |