# **Technical Document**

## **Synthentic Financial Manager System For Fraud Transactions**

## **Juan Guillermo Rute Sanabria**

Contenido

[Introduction 2](#_Toc511377859)

[Requeriments 2](#_Toc511377860)

[Technologies 2](#_Toc511377861)

[General Diagram 3](#_Toc511377862)

[Class Diagram 4](#_Toc511377863)

[DataBase 7](#_Toc511377864)

[Model Entity Relationship 7](#_Toc511377865)

[Stored Procedure 8](#_Toc511377866)

[Unit testing 9](#_Toc511377867)

|  |  |  |
| --- | --- | --- |
| Version | Author | Date |
| V1 | Juan Guillermo Rute Sanabria | April 11,2018 |
|  |  |  |

# Introduction

This document allow us to show the important decisions taken for the implementation of the application, are included the user manual and configuration manual.

We can get information about how was implemented the web application build with the architectural pattern Model-View-Controller(MVC) to allow user create transactions, mark a transaction as a fraud and search it. Also we can get information about the service component where is exposed the functionalities of register bank transactions and search transaction returning XML message.

# Requeriments

Based on the requirements document [NET Test v1.2.pdf](NET%20Test%20v1.2.pdf) the principal goals are:

|  |  |
| --- | --- |
| Description | Status |
| web application with MVC | Done |
| Allow a user register transactions | Done |
| Mark a transaction as a fraud | Done |
| Search transactions | Done |
| Authenticatication and roles access. | ToDo |
| Web Service to perform Assistant role operations (register and search). | Done |
| Implement a relational database design | Done |

Time to resolution: Start date April 11, 2018 (10 AM) and end date April 13, 2018 ( 11 AM)

# Technologies

In the table below shown the principal tools and technologies used in the implementation.

|  |  |
| --- | --- |
| **Tool or Technology** | **Description** |
| Microsoft Visual Studio Professional 2017 | Development Environment |
| .NET Framework 4.6.1 | Framework |
| C# | Main programing language used for the development. |
| ASP.NET Web Application | Template of .Net Framework using MVC |
| Git Desktop V 1.1.1 | Tool to our repositories od source code. |
| SQL Server 14.0.17 | DataBase server using the SQLEXPRESS version. |
| Bootstrap | Front-end component library to build responsive pages |
| Razor | Markup syntax that allow us embed c# code in the views |
| Unity | Dependency injection container |

# General Diagram

**FraudTransactionsJuanRute**: is an ASP Net Web application using MVC template, contain the crud functionality that allows the user create new transactions, list all the transactions and apply filters on that.

**DataAccessLayer**: Is a dynamic link library that encapsulate the communication to the database through Entity Framework.

* WebStuff, has a façade with all the operations needed for the controllers of the web application using EntityFramework to access to the tables.
* ServiceStuf, has a façade with all the operations needed for the controllers of the service in this case using a store procedure.

**RestAssistantService**: Wep Api application that expose all the operations used by the assistant role.

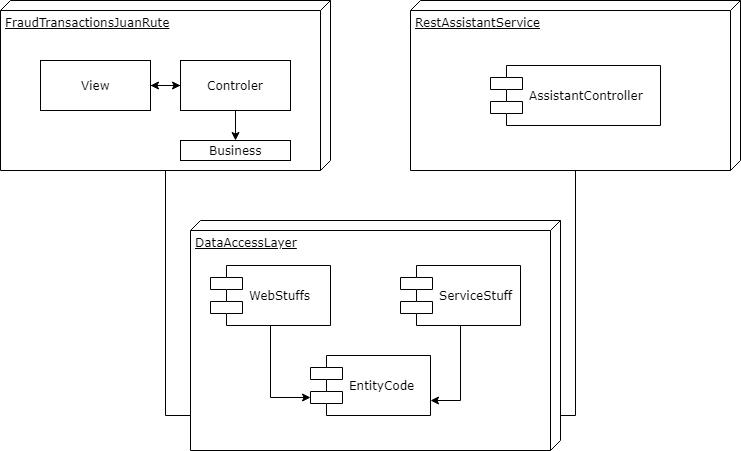


Ilustración 1General Diagram

## Class Diagram

**DataAccessLayer ServiceStuff:** Interfaces and classes used for the service**:**

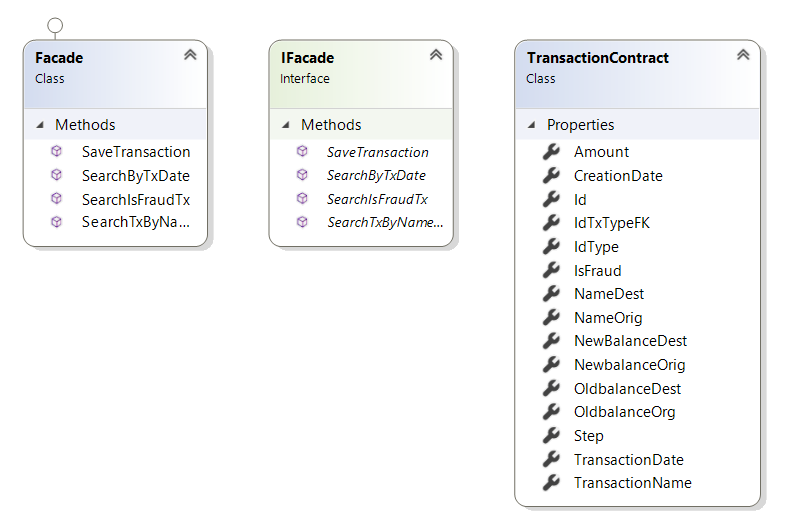


Ilustración 4ServiceStuff Classes

**DataAccessLayer WebStuffs:** Interfaces and classes used for the Web Application**:**

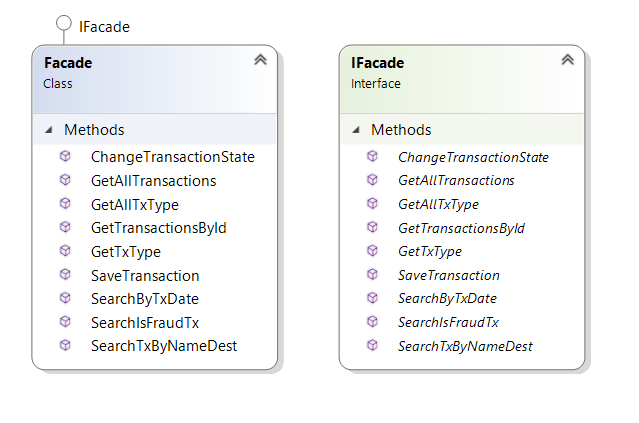


Ilustración 6WebStuff Classes

**DataAccessLayer EntityCode** Classes to perform the request to sql database through entity framework:

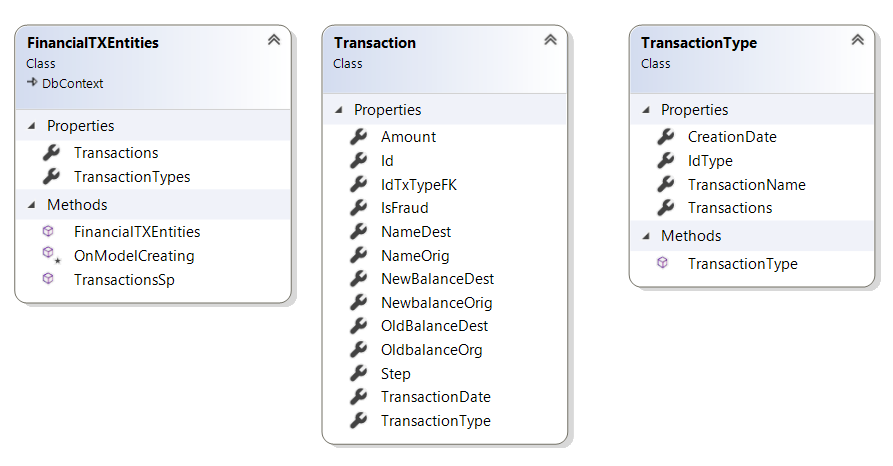


Ilustración 7 EntityCode Classes

**FraudTransactionsJuanRute** controllers and business classes to perform the request in the MVC web app:

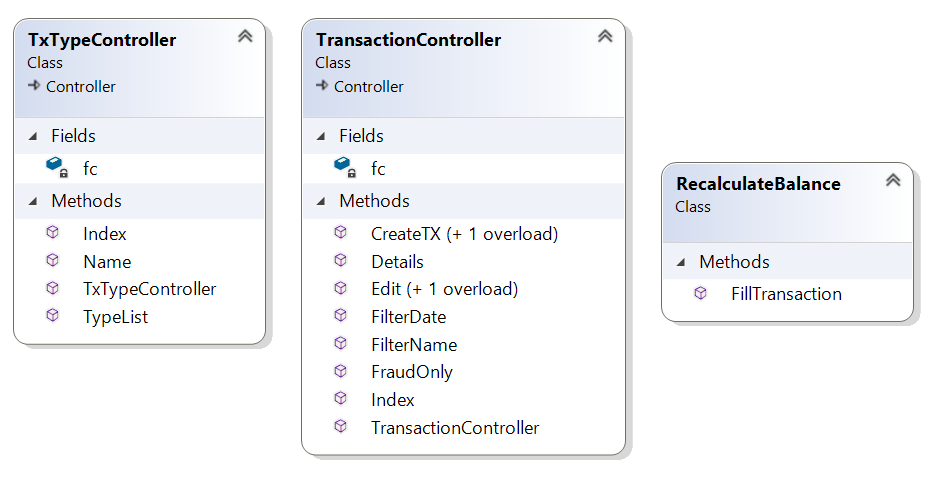


Ilustración 5 MVC App clases

**RestAssistantService,** classes in the web service project

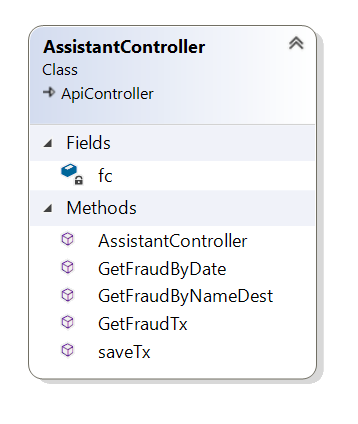


Ilustración 8Web Api class

# DataBase

## Model Entity Relationship

Using SQL Server Management Studio was create the data base diagram bellow:

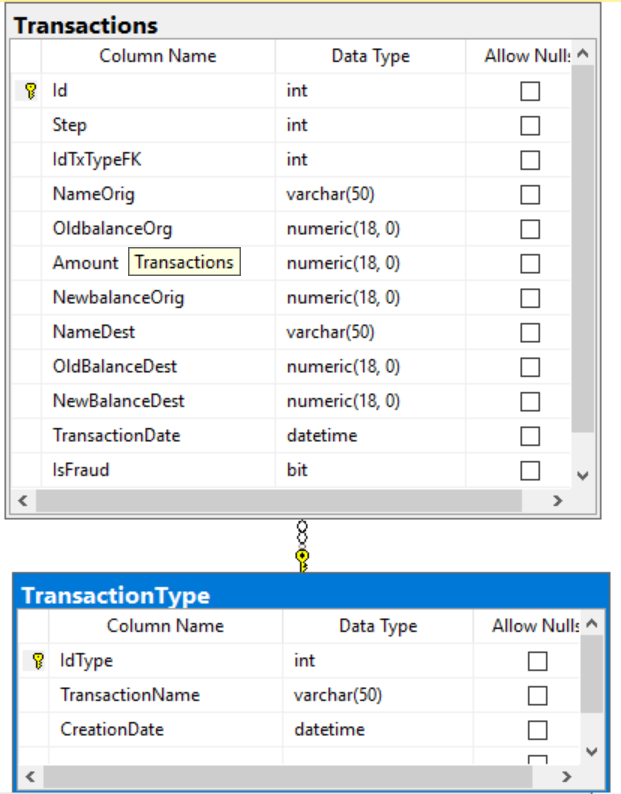


Ilustración 2Entity Relationship

In SQL Server 2017 we can show the object tree as the image bellow:

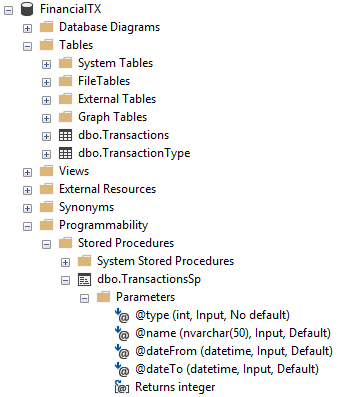
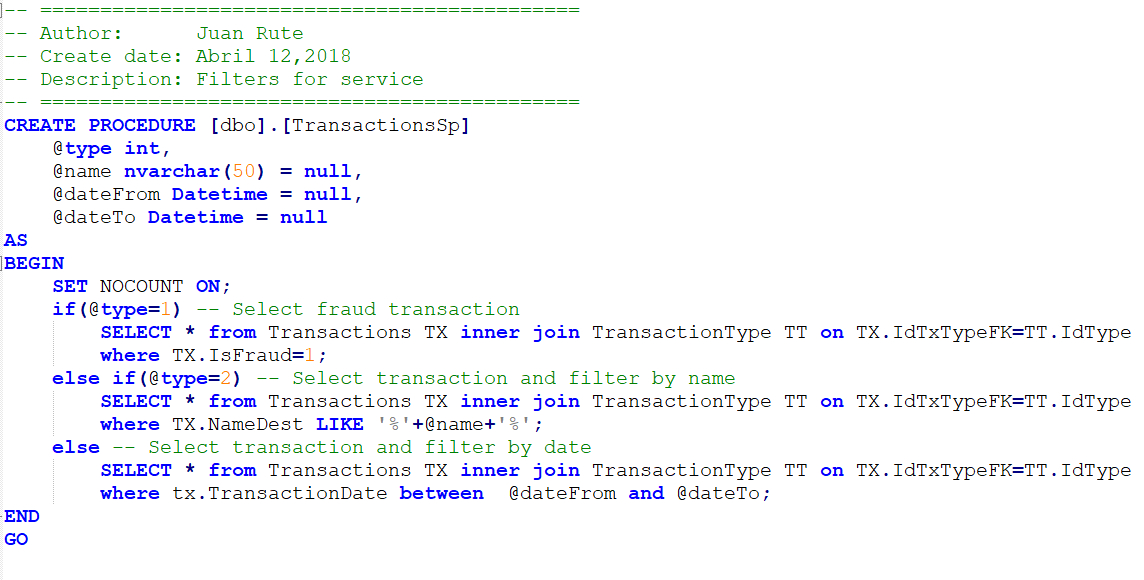


Ilustración 3 Objects in database

## Stored Procedure

Were created different tables and a stored procedure called TransactionsSp to manage all the methods of the service. Depending on the type parameter we can define which query will execute:



# Unit testing

For unit testing was created the project ‘FraudTransactionsJuanRute.Tests’ in this was written a test for ‘FillTransaction’ method, this method complete the information of a transaction

