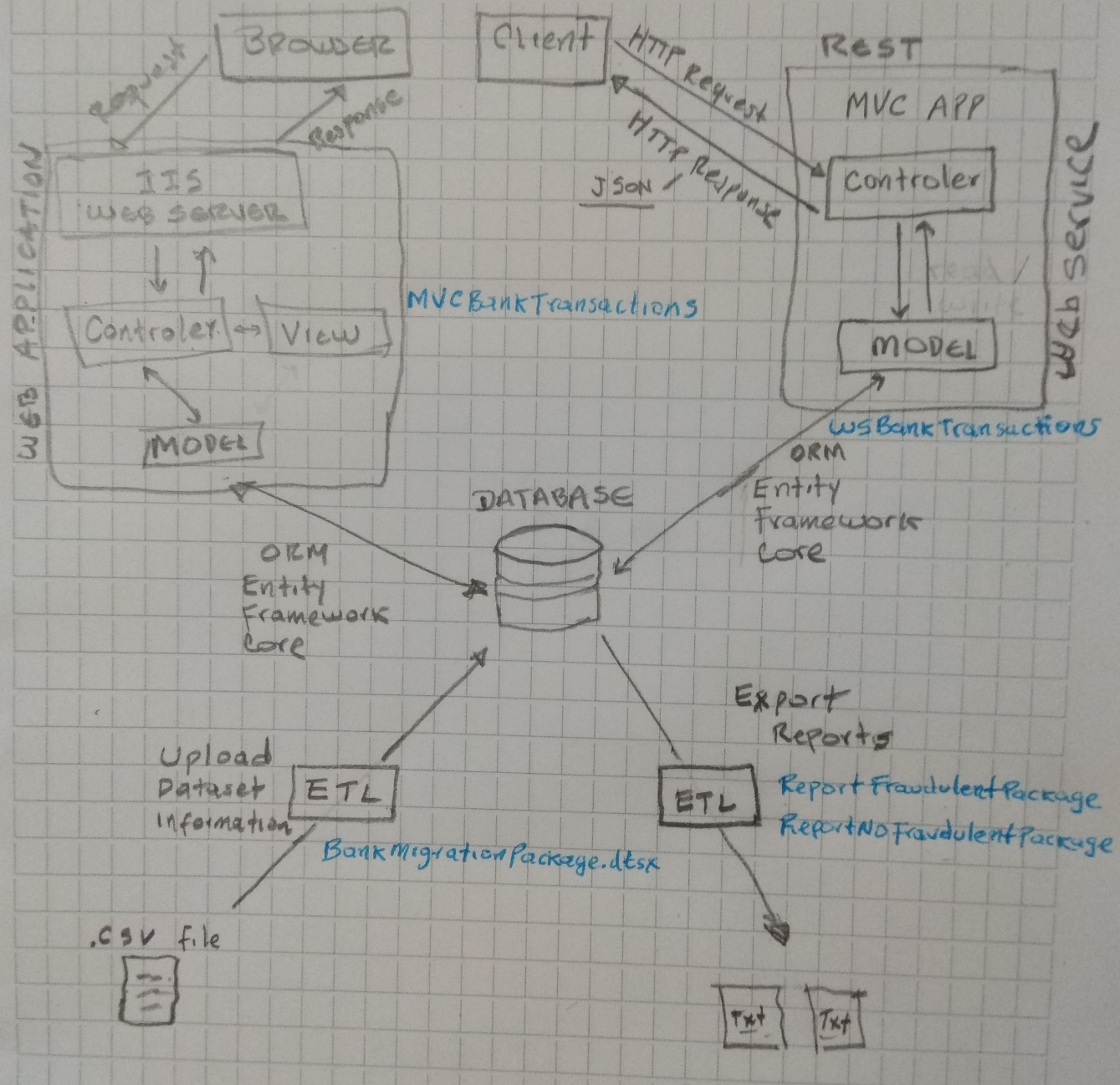
**DESCRIPCION DE LA SOLUCION**

En este pequeño documento pretendo presentar las generalidades de la solución planteada al ejercicio expuesto.

Diagrama de arquitectura general



En el anterior diagrama se muestra la arquitectura usada para la solución, donde se muestran los componentes principales:

**MVCBankTransactions**: ASP.NET Core Web Application (.NET Core) con funcionalidad de registrar, consutar, modificar, eliminar transacciones. Usa Entity Framework para implementar la funcionalidad de manejo e interaccion con la base de datos.

**WSBankTransactions**: web API with ASP.NET Core para implementar el web service que registra y consulta transacciones. Usa Entity Framework para implementar la funcionalidad de manejo e interaccion con la base de datos.

**BankMigrationPackage.dtsx**: ETL creada con SQL SERVER 2016, usando la funcionalidad Import and Export Data para su creación. Toma la información contenida en el archivo PS\_20174392719\_1491204439457\_log.csv y la sube a la base de datos en la tabla tmpTransactions.

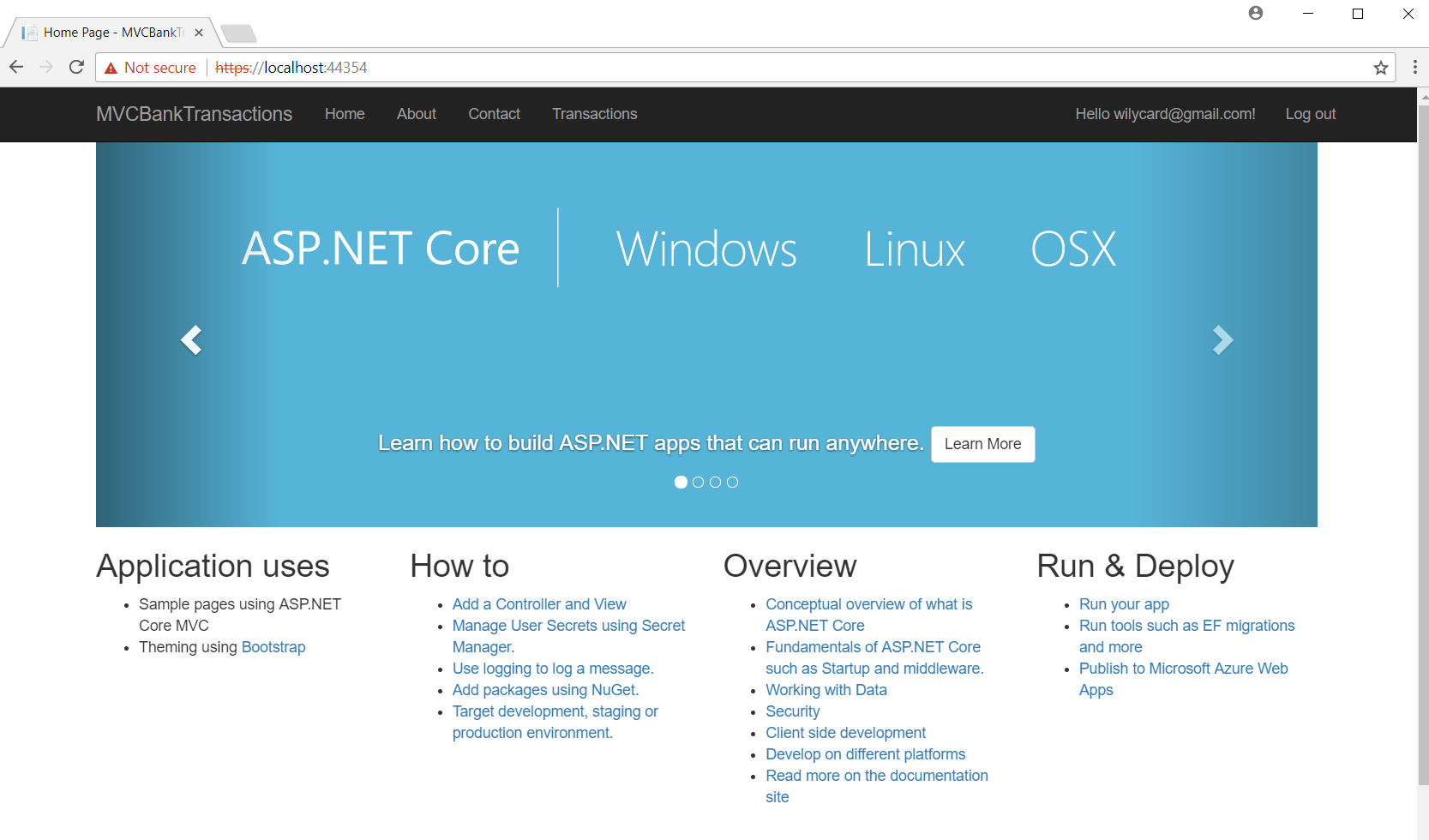
**ReportFraudulentPackage.dtsx**: ETL creada con SQL SERVER 2016, usando la funcionalidad Import and Export Data para su creación. Genera reporte de las transacciones marcadas como fraudulentas y lo exporta al archivo plano BankReportFraudulent.txt.

**ReportNoFraudulentPackage.dtsx**: ETL creada con SQL SERVER 2016, usando la funcionalidad Import and Export Data para su creación. Genera reporte de las transacciones que no son fraudulentas y lo exporta al archivo plano BankReportNoFraudulent.txt.

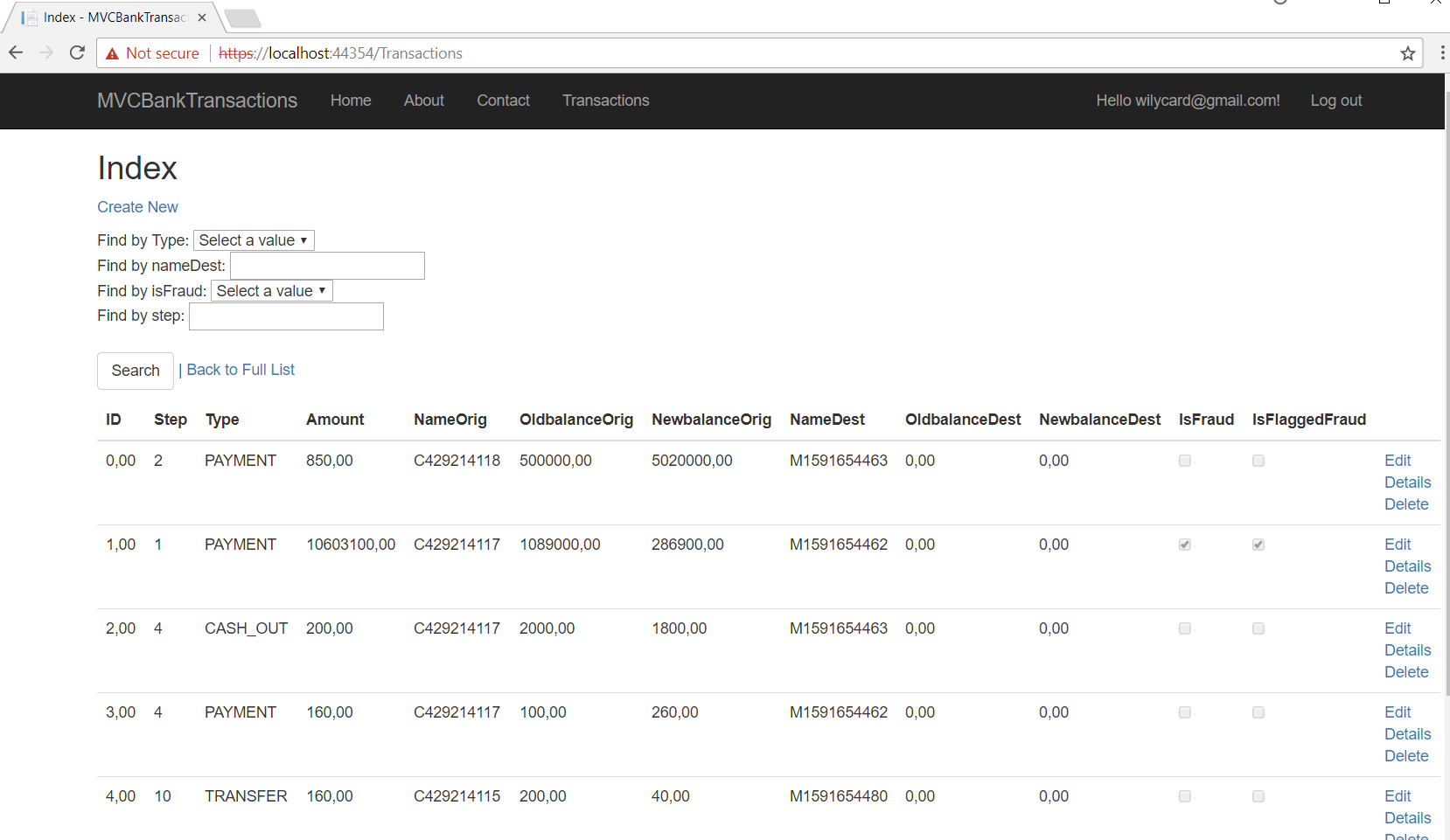
BankTransactions: Base de datos creada sobre SQL Server 2016, creada usando Entity Framework desde el modelo.

**CAPTURA DE PANTALLAS DE LA APLICACIÓN WEB**

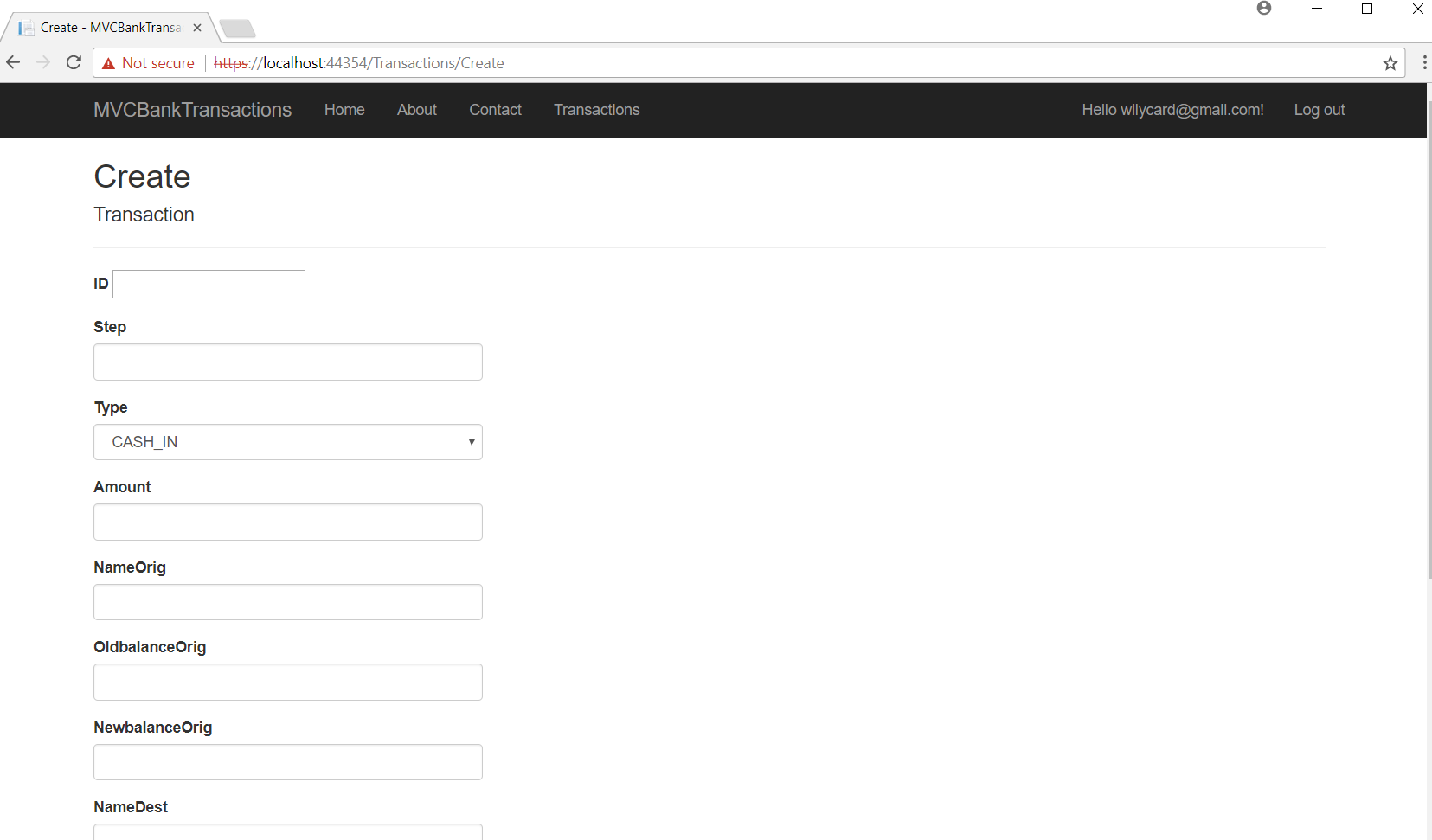
Pantalla de inicio:



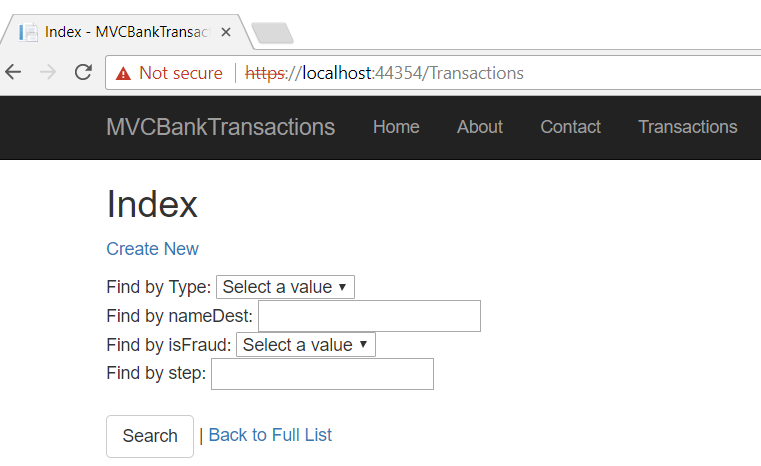
Listado de transacciones: Se pueden editar, ver detalle y eliminar



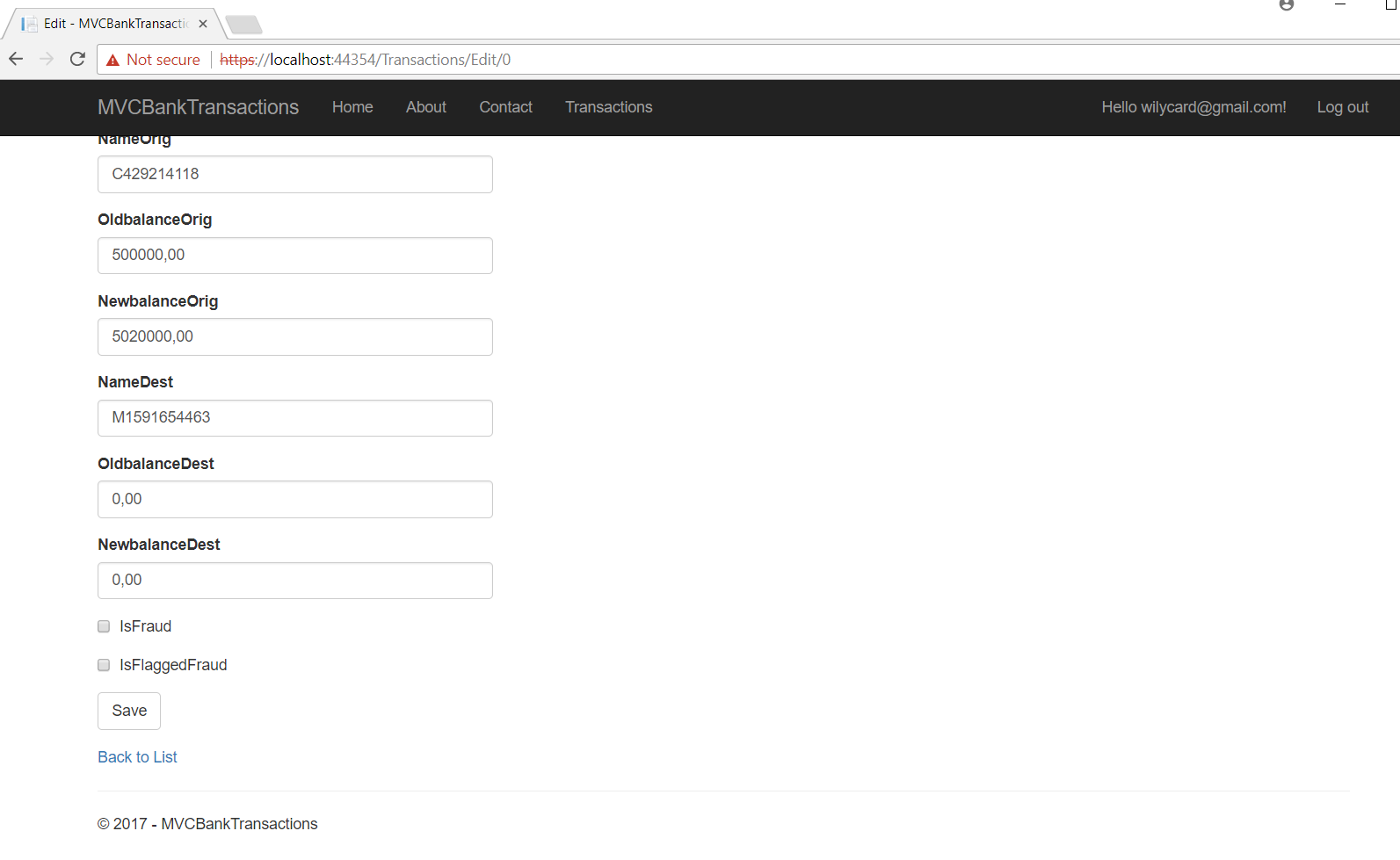
Registrar Nueva Transacción



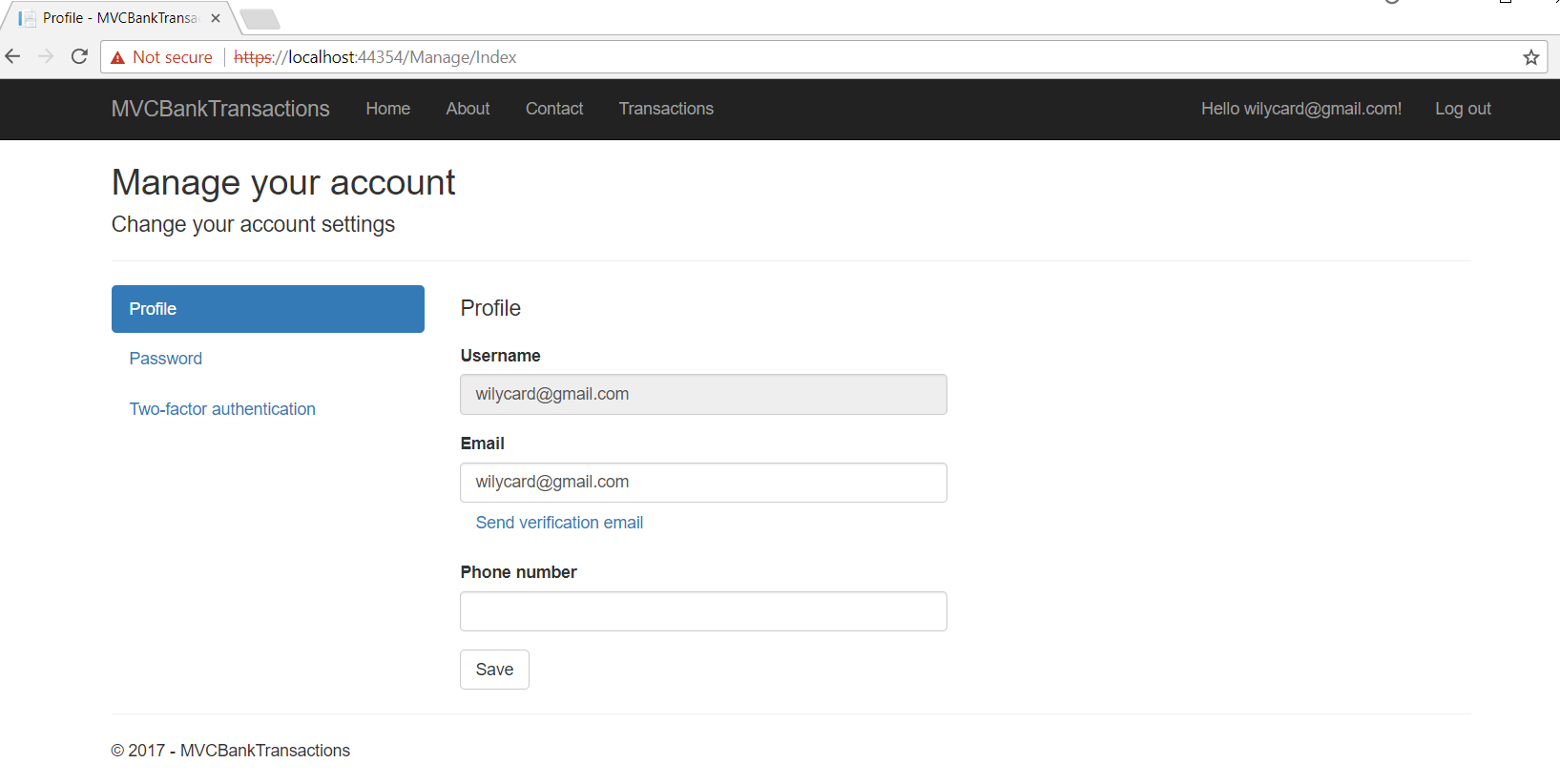
Consulta de transacciones por los diferentes criterios de búsqueda:



Opción de editar, donde se pueden marcar como fraude o modificar cualquier campo:

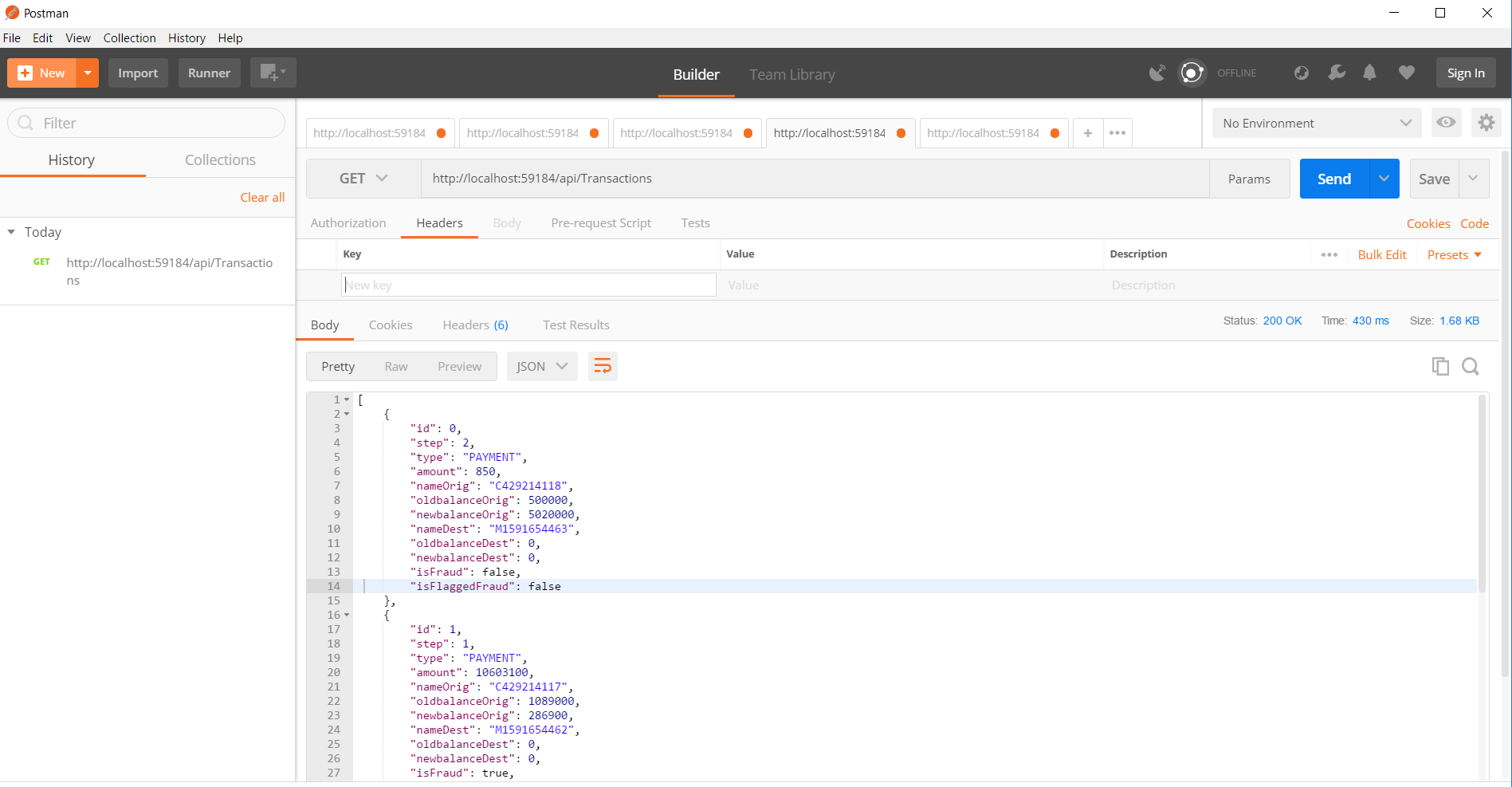


Se implementó modelo de seguridad para los usuarios:

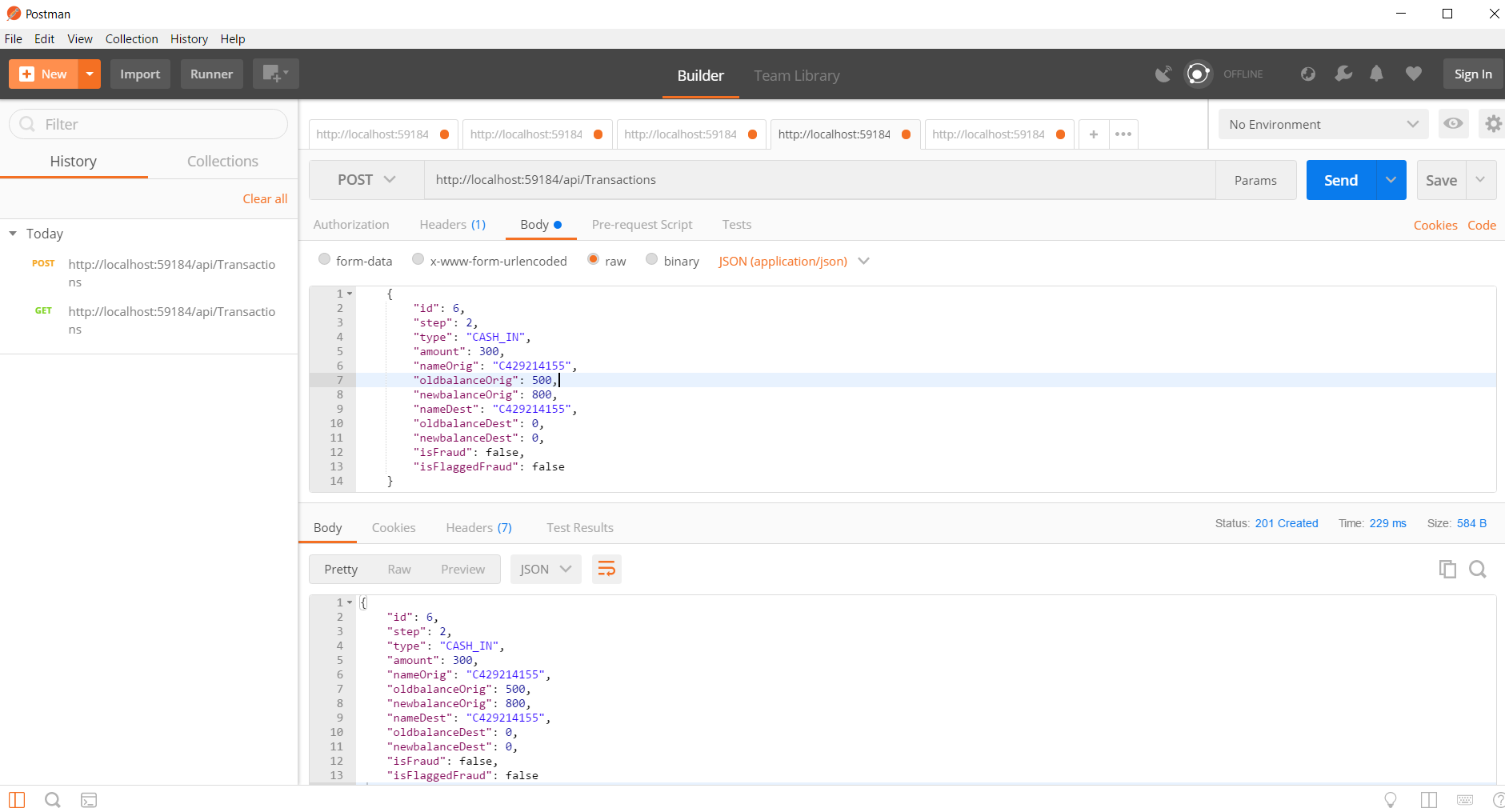


**CAPTURA DE PANTALLAS DEL FUNCIONAMIENTO DEL WEB SERVICE**

GET Transactions:



POST Transaction:



Consulta en la base de datos:

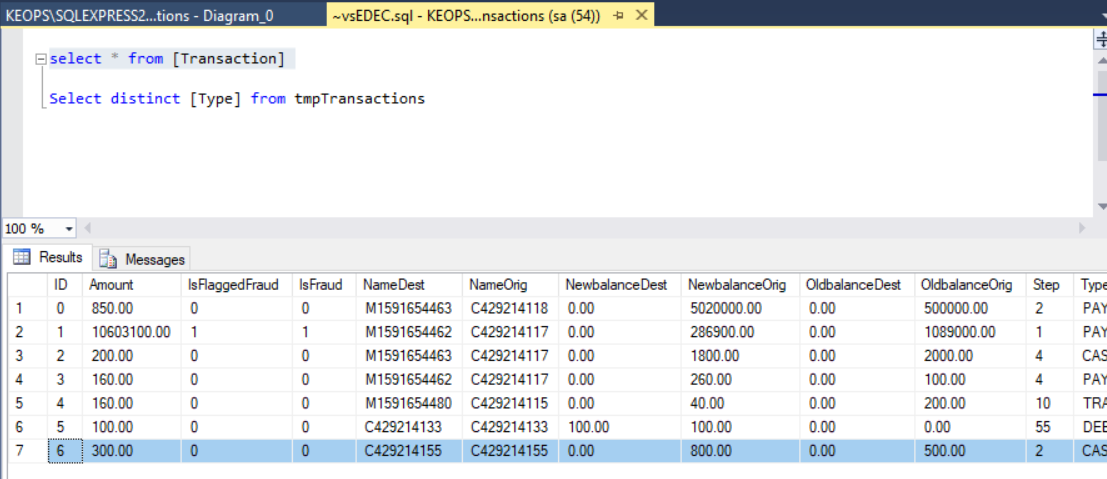


DIAGRAMA DE LA BASE DE DATOS: En el diagrama se muestra la estructura creada para la implementación de un esquema de seguridad Role based Authorization.

