# Penny’s Payment Problem

## Outline

The payments team at 8digital consists of Penny, who carries out a manual procedure to handle payments when customers make an order.

8Digital with it's newly re-branded api is hoping to dramatically increase the amount of music they sell so your team has been formed to try and automate the process.

This is what currently happens:

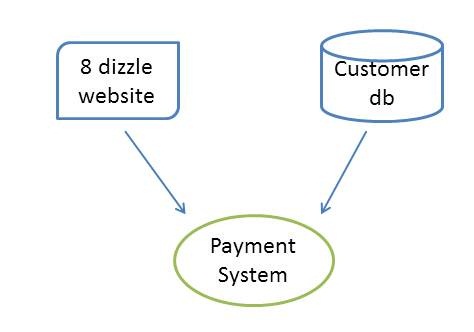
*Penny gets an email with the customers order and credit card details. She then pastes it into a spreadsheet, types the transaction into a POS terminal and the ticks the spreadsheet to say she's taken the payment.*

*She then opens Outlook and creates a new Receipt email using a template (she's that advanced) fills in the Customers details and purchase details and clicks send.*

*She then updates the spreadsheet again.*

*Finally she fulfils the order, by sending Lauren an email asking her to add the track into the customers locker and finally updating the spreadsheet.*

**Your objective is to design this system, which will integrate with a separately developed web application, and database containing user information.**

****

**Website**

Within the development team it has been agreed that the website application will need to change to accommodate the new payment system. Therefore, you have the freedom to decide how the two will interface and communicate.

The website is an ASP.NET MVC 2 application.

**Customer db**

Inside here all a users details are stored such as name, address, email address. Data for orders made on the website, and from the legacy system have already been imported into to this database. There is capacity for data from the payments system to reside here also, if required.

This is a SQL Server database running on a dedicated server.

## Functional Requirements

* The system should integrate with the following payment providers
  + Paypal
  + Google checkout
  + “Chez” – handles all kinds of credit cards
* The billing department of the company need to be able to view the data for the past 5 years and perform complex pre-defined queries such as the following (they have no SQL skills)
  + How many orders in the last n days/weeks/months
  + View orders from a certain customer, or subset of customers (e.g. bycountry)
  + Must be easy to import into excel
* The customer services team also need to be able to
  + View a single customer’s payment and refund history
  + Automatically initiate refunds
  + Push payments through manually
* A weekly email summarising some key reports (for the previous week’s data) needs to be emailed to the billing, sales and customer service departments on a Monday morning. Without this information to submit the company will be fined
* An email needs to be sent to customers informing them when a payment or refund has been processed
* All payments (and refunds etc) should be available for reporting within 24 hours of occurring
* All data older than 5 years needs to be made available within 3 days of being requested

## Non-functional Requirements

These are some but not all non-functional requirements

**Reliability**

* The company cannot lose any of this data

**Performance**

* Confirmation emails need to be sent within 1 hour of a payment or refund request being received (allowing payment provider 15 minutes to do their bit)
* None of the billing/customer services searches may take more than 10 seconds to be fully rendered

**Security**

* Data protection legislation
* Prevent fraudulent transactions

**Scalability**

* The company are rapidly expanding at a rate of 1000 extra payments a month and expect this to continue for the next 6 months. They are not sure what will happen beyond that

**Localisation & Internationalisation**

* Needs to handle different languages
* Will also need to handle different currencies
* Internal reporting services can all be English

**Availability**

* The business just wants to avoid losing money, and considers inability to take payments as a loss of earnings
* Downtime of the internal applications is less damaging, but can still be costly if the data is required urgently