



Sample

Dissertation #4

(Front Matter removed)

Table of Contents

1.	<i>Introduction</i>	9
1.1	Overview	9
1.2	Technical Architecture	9
1.3	Project Objectives.....	10
1.4	Project Challenges.....	10
2.	<i>Research</i>	11
2.1	Background Research	11
2.2	Alternative Existing Solutions	12
2.3	Technologies Researched.....	15
2.3.1	Frontend Technologies Researched.....	15
2.3.2	Backend Technologies Researched..	16
2.4	Other Relevant Research Done	17
2.5	Resultant Findings and Requirements	21
3.	<i>Design</i>	23
3.1	Overview	23
3.2	Approach and methodology	23
3.3	Technical Diagrams.....	25
3.4	User experience and interface	31
4.	<i>Implementation and development</i>	32
4.1	Detailed System Overview	32
4.2	Development Process	45
5.	<i>System validation</i>	49
5.1	Manual testing	49
5.1.1	Functional testing.....	49
5.1.2	Non-Functional testing	50
5.1.3	Test Plan	51
5.2	Automated testing	53

6.	<i>Issues and risks</i>	54
6.1	Issues.....	54
6.2	Risks	55
7.	<i>Project plan</i>	56
7.1	Future Work.....	56
7.2	Other Features	58
8.	<i>Conclusion</i>	59
9.	<i>Bibliography</i>	60
10.	<i>Appendix</i>	63
a.	User Requirements Survey.....	63
b.	UAT Complete Survey.....	67
c.	Automated Test Results.....	70
d.	Google Cloud Costs.....	71
e.	Time Series Javascript	72

Table of Figures

Figure 1 Architecture Diagram.....	9
Figure 2 ICRTouch Touchpoint.....	12
Figure 3 Vend Retail.....	13
Figure 4 Barcode Scanner.....	17
Figure 5 EAN/UPC Barcode.....	17
Figure 6 Hanspree Touchscreen.....	17
Figure 7 Hashing SQL.....	18
Figure 8 Sample Software Patent.....	19
Figure 9 Live Drawer.....	20
Figure 10 FDD Example.....	23
Figure 11 Waterfall Model.....	24
Figure 12 Agile Scrum.....	24
Figure 13 Architecture Diagram.....	25
Figure 14 ERD Diagram.....	26
Figure 15 Use Case Diagram.....	28
Figure 16 Sequence Diagram.....	29
Figure 17 Class Diagram.....	30
Figure 18 Peter Mowile's UX Honeycomb.....	31
Figure 19 Local Application.....	32
Figure 20 NetBeans Project Layout.....	35
Figure 21 Web Application possolutions.ie.....	36
Figure 22 Visual Studio Project Layout.....	40
Figure 23 Cloud Server.....	41
Figure 24 DNS Settings.....	41
Figure 25 Cloud SQL.....	42
Figure 26 Github master branch.....	43
Figure 27 Github local branch.....	43
Figure 28 Trello Kanban Board.....	44
Figure 29 Simplified code of change Algorithm.....	45
Figure 30 SMTP Server code snippet.....	46

Figure 31 File Server Upload Feature.....	47
Figure 32 Connecting to the File Server.....	47
Figure 33 Detecting File Extension.....	48
Figure 34 Initial Test Results.....	53
Figure 35 Receipt Lines Table Representation.....	54
Figure 36 Parsing Script.....	55
Figure 37 Azure Machine Learning Model.....	56
Figure 38 Linear Regression Example.....	57
Figure 39 Cost Function Linear Regression.....	58

1. Introduction

1.1 Project Overview

This project is a hybrid system, comprising of a local Electronic point of sales system and a cloud-based management system. RetailOnline is the name of the system I have developed. All data syncs to the cloud every two minutes ensuring the most relevant data is available at your fingertips. The local system has an in-built database so even if there is no internet connection available, sales are still recorded offline. All users are managed by different levels of roles and permissions, so all security concerns are covered.

This project was inspired by my keen interest in enterprise systems - particularly retail. I am a part-time supervisor in Penneys, this is where the spark for my idea came from. I contemplated how management and sales technologies could be improved. Experience that I have gained during my internship and my current knowledge of programming from my last few years study, greatly benefited me throughout this project.

1.2 Technical Architecture

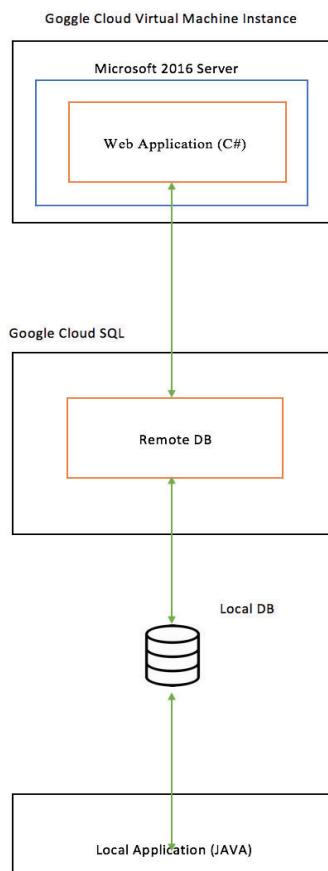


Figure 1 Architecture Diagram

1.3 Project Objectives

The overall end objective of this project is to have a fully functioning EPOS system, with a website that can fully control point of sales from anywhere, displaying sales data elegantly and predicting future sales so the correct amount of stock is in store at all times. This is done by achieving the following objectives:

- Develop a fully functioning stand-alone java application to run on a till (EPOS). Which records all stock / cash in and out of the shop on a particular day.
- Develop a full backend with a webserver, database, mail server and predictive analysis to make meaningful decisions on stock ordering.
- Develop all the backend services / microservices in the cloud- allowing the local application to access the services 24/7.

1.4 Project Challenges

The project developer must first become familiar with a new technology stack to carry out the proposed development of this application, which took a considerable amount of time to learn.

Another issue is testing the system, as the systems functionality is broad a lot of dummy historical data was needed. As it's a college project the user base was small so for testing about ten different dummy users were created with different roles and permissions to make sure every aspect of the system is working correctly.

An analysis of the functional requirements shows that most of the features proposed have their own complexity and problems that need to be addressed. For example, implementing the SMTP server so that orders can be emailed directly to the suppliers warehouse for picking.

2. Research

2.1 Background Research

Introduction

Before I could start this project, a number of aspects of my project had to be researched. Since this is not like other web or local based EPOS systems, my system is a hybrid of both.

I talked to friends and peers who work for different retail companies, there feedback was helpful, and I discovered that a lot of companies are still using out dated software, that is running on their own local server instore. This is mainly due to the fact that a lot of the senior people in these companies believe in the approach of if it is not broken don't fix it. I think if the efficiency and productivity benefits of the new software were demonstrated to them, they may consider updating their software within their business.

The use of Google Scholar was very helpful during this project, I found some documents which were helpful to develop basic requirements. These documents were used for my background research. (1)

How do IT Systems help business / trading ?

"The advent of information technologies has had a huge impact on how organizations design, implement, and support business processes. From document management systems to ERP systems, information systems are tied into organizational processes. Using business process management, organizations can empower employees and leverage their processes for competitive advantage. Using business process reengineering, organizations can vastly improve their effectiveness and the quality of their products and services. Integrating information technology with business processes is one way that information systems can bring an organization lasting competitive advantage." - Harvard Business Review 68.4 (1990): 104–112 (2)

The research for this project took longer than expected. It was ongoing throughout the project, because this was the only way to ensure the most functionality could be incorporated. The main technologies/ frameworks were decided at the start, but research didn't stop until a few weeks before the end of the project.

There are many EPOS systems on the market at present, the following are some of the features offered by two such products. These two systems are aimed at a similar market to the software developed in this project.

2.2 Alternative Existing Solutions

ICR Touch

Founded in 1999 ICR Touch are an EPOS software development company based in England, they specialise in front of house EPOS software for the retail and hospitality sector. For this report I am focusing on their retail software which is called “TouchPoint”.

“TouchPoint software is more than a decade in development and has sold more than 70,000 licences worldwide” (3)

- Customisable on look, layout and functionality
- In built product promotions
- Secure with a range of operator sign on options
- Multi lingual, multi currency
- Runs on Windows OS

(4)

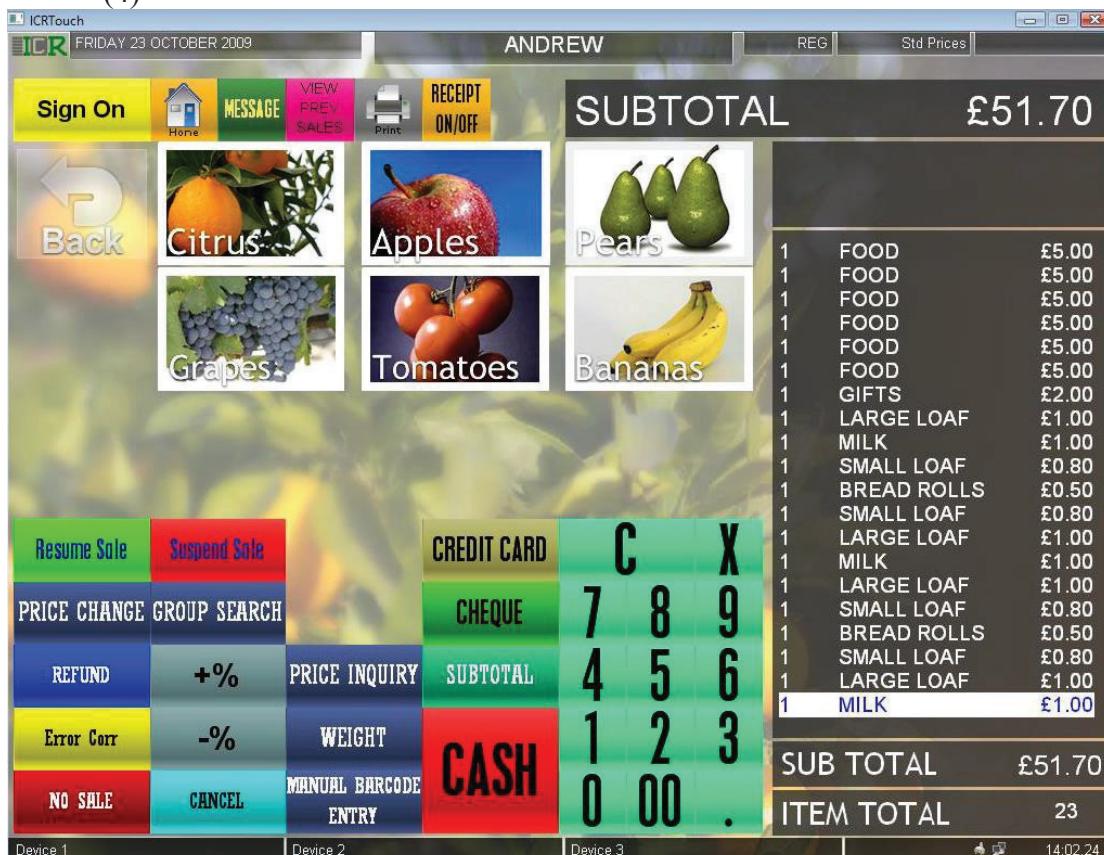


Figure 2 ICRTouch Touchpoint

Vend

Vend is a relatively new Canadian EPOS software development company. They believe in cloud-based EPOS as the essential solution for retailers. Their full system runs on the cloud, allowing users to log in from anywhere.

It is an EPOS system, but I personally think it lacks in features. Its more focused on the reports for products and sales which are important, but I think the focus should be more on transactions and point of sales. First get that right, then the reports will follow.

“Break free from the counter, stockroom or office. Vend is cloud-based POS software, so you can sign in and work from anywhere. Easily access your sales, products and reports, and always have the latest data.” (5)

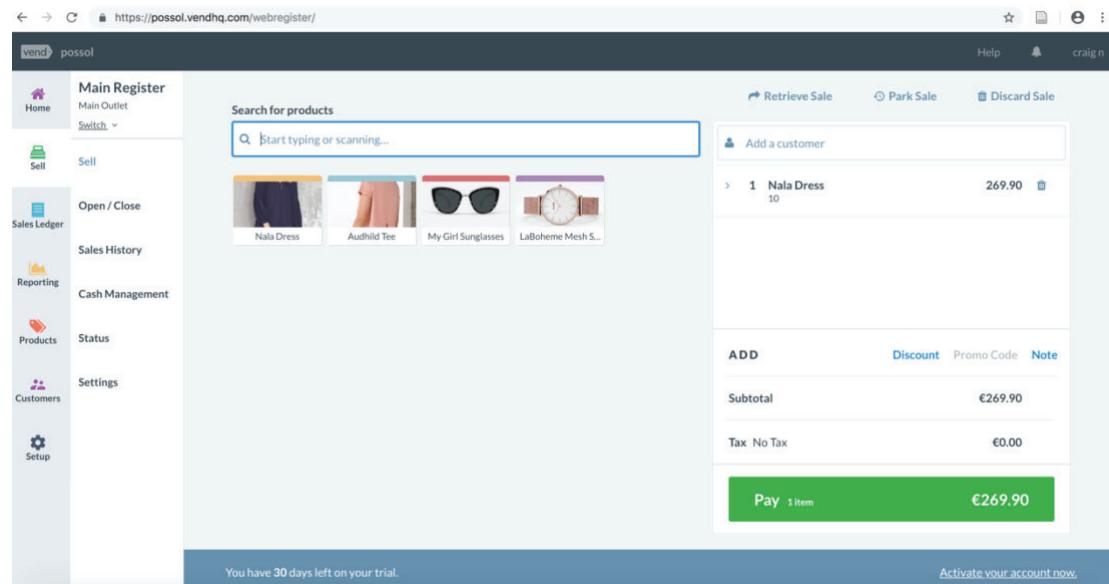


Figure 3 Vend Retail

Comparison – Difference between the two researched systems and RetailOnline

Product	Local System	Remote Server	Automatic Sync	Reports	File Server / Storage	Till Totals / Automatic	User Permissions	Multi-platform
ICR Touch	✓			✓			➢➢➢	Windows
Vend		✓		✓		✓	✓	Specific Platforms
RetailOnline			✓					

2.3 Technologies Researched

In RetailOnline, the primary method of user input is through the local application. The application stores the relevant user's information and information related to any actions they perform. Research was carried out, as part of this it was decided to use the technologies below. Due to their strengths for this particular project.

2.3.1 Backend

For the backend, I researched a number of different frameworks which alleviates the time-consuming work, like querying the database so the application can focus on its main functionality.

C# & .NET

C# is a general-purpose object orientated programming language, developed by Microsoft in 2000.

.NET is the framework that supports C# as Microsoft describes it as a free, cross-platform, open source platform for developers building anything from web based applications to games. (6)

Database

A number of different Databases were researched including Neo4J, SQLlite, MySQL, PostgreSQL and MongoDB. Each have their own benefits for different types of systems, but I decided that a relational database would be the best for my project. It provides clear structure for storing data, as orders and receipts have clear structure so does a relational database, so SQL is the preferred choice.

MySQL

MySQL is an open source RDBMS (relational database management system) owned by Oracle (sun microsystems) but originated from Sweden AB in 1995. Michael Widenius was the main developer of the original version of MySQL. It uses the well-known LAMP stack (Linux, Apache, MySQL, PHP), which many web based systems use.(7)

Google Cloud Platform

This is a platform for running apps, websites and storing data in the cloud. It is scalable which means future expansion is very simple. Also google provides security with cloud instances that keeps code, data and backup files safe. It also allows easy collaboration between users.

Microsoft 2016 Server

Google cloud console allows a user to run a server on a virtual machine. I researched different types of servers, in terms of cost effectiveness and low processing time, Microsoft servers are the fastest. An instance of a virtual machine can be ran using Google cloud.

Cloud SQL

Part of the google cloud platform is Cloud SQL, which allows the hosting of an SQL database in the cloud. User permissions and IP address access can be set on the SQL instance while it is running.

2.3.2 Frontend

For the frontend, a number of frameworks / languages were researched. I require some code to be executed on the client side to make the web app more responsive, also this will prevent server overload on the backend.

Java

Java is also an OO programming language. The java project was launched in 1991 with James Gosling as the lead developer. The basis to java was write once, run everywhere and this has been Oracle (sun microsystems) mission ever since. Unlike other languages java does not need to recompile it runs on a java virtual machine also referred to as JVM. (8)

Bootstrap

Created in 2012, bootstrap is a free open source front end framework for designing professional looking websites and webapps, it comprises of HTML, CSS and JavaScript. It's suitable for both mobile and desktop development and it makes use of available screen space automatically, depending on the screen size of the device. (9)

CSS

CSS is mainly used for front end presentation. It is used for the presentation of a document written in a mark-up language. CSS is designed to separate the presentation and content, having this separation allows accessibility, better interoperation and generally looks neater for the incoming user.

HTML

Founded in 1980 by Tim Berners-Lee, HTML is the standard mark-up language for creating webpages. Web browsers fetch HTML documents from local storage or Web browsers and the browser then renders these documents. (10)

2.4 Other Relevant Research Done

Barcode Scanner

A barcode scanner also referred to as a barcode reader by many big companies, comprises of a light source (usually red laser), a sensor/lens and a decoder as standard. It takes in an analogue signal, the black and white barcode usually printed on the label of the item. This is then converted into a digital signal, then this signal passes through the decoder, which sends the barcode as ASCII text to the computer the scanner is connected to. (11)

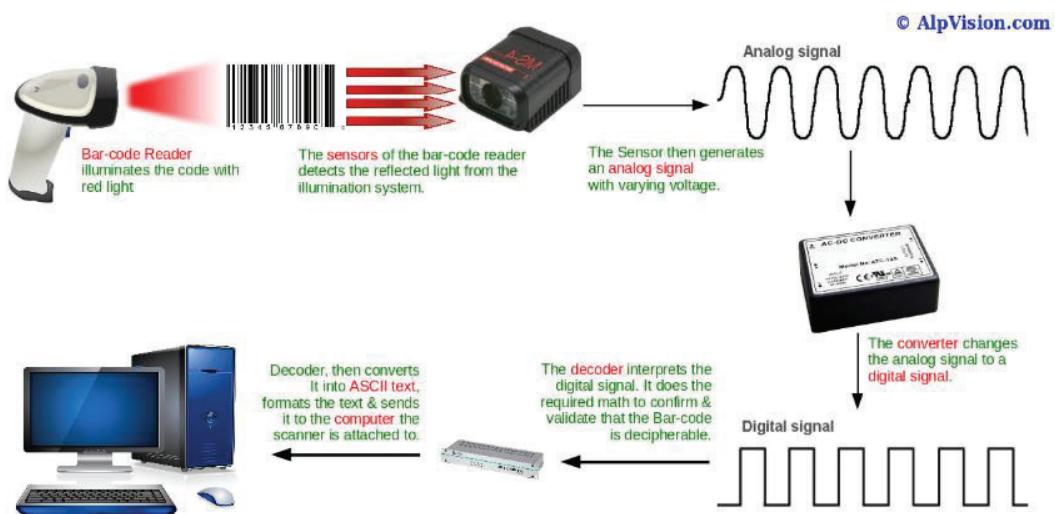


Figure 4 Barcode Scanner

Barcode Types

There are many different types of barcodes including EAN, UPC, PDF417, Interleaved and in most recent years QR Codes (mainly used for mobile devices). But in retail the most common is a UPC (universal product code) / EAN. Which is the 13-digit unique code found on the side of the product. All numbers are assigned by GS1 – the global standards organisation. (12)



Figure 5 EAN/UPC Barcode

- Omnidirectional – meaning it can receive signals on all directions from the barcode scanner. It does not matter on the angle or the exact distance the barcode is from the scanner.

Touch Screen

There are many different types of touch screens on the market today. But from a business perspective you want one that's fast, low- latency and multitouch from all directions. Throughout this project I researched multiple different touch screens from low end to high end ones. I settled for the following one because it had all the features required and it supports both HDMI and VGA, meaning it will work with a range of computers not limiting itself specifically to windows OS.



Figure 6 Hannspree TouchScreen

(13)

The above touch screen is ideal for a start-up retail store. Its low cost, has tempered glass on the screen and supports 10 point multitouch so it will respond fast to the cashiers demands.

Password Security

A significant problem in every aspect of software design now is security and privacy. Passwords can no longer be stored as strings anywhere in the system because this increases the chances of an SQL injection attack. So instead the most reliable solution is to store the passwords as hashes in the database.

SQL provides many different inbuilt functions for hashing including MD5, SHA1, SHA2, AES and many more. MD5 is no longer reliable as it is not a strong encryption method and can be easily hacked by someone who has sufficient expertise. Instead it's safer to use SHA2 which lets you pass in 2 parameters, the first one is the plain text to be hashed the second is the bit length of the hashed result. This must have a value of (224,256,384,512), obviously the higher the bit the more secure the password will be. The function then returns the desired hashed value in hexadecimal format, which can then be stored in the database. (14)

```
mysql> SELECT SHA2('abc', 224);
-> '23097d223405d8228642a477bda255b32aadbc4bda0b3f7e36c9da7'
```

Figure 7 Hashing SQL

Patents

“Patents protect technological advances. New or improved products or processes are eligible for patents. A patent confers upon its holder, for a limited period, the right to exclude others from exploiting (making, using, selling, importing) the patented invention, except with the consent of the owner of the patent. A patent is a form of ‘industrial property’, which can be assigned, transferred, licensed or used by the owner. Patents are territorial and give an exclusive right in the country where the patent has been granted as long as the patent is renewed each year through the payment of a renewal fee.” – Irish Patents Office 2019 (15)

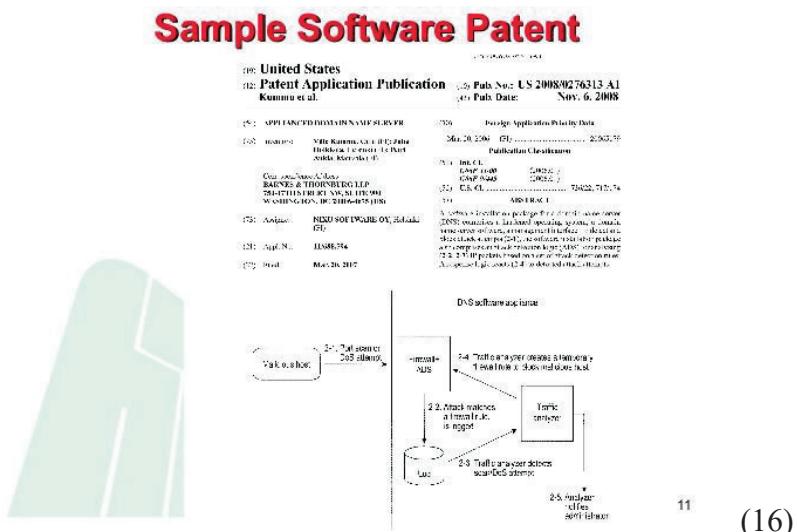


Figure 8 Sample Software Patent

From extensive research, I have concluded that software patents are a lot more difficult to ascertain than normal patents. There has to be proof that the idea/concept is a new revolution and not something that is already in existence.

Therefore, for this project I decided not to apply for a patent, but it is good knowledge to have for the future. I plan on adding more features to this project after college and increasing the scope of the system. In the future work section at the end of this report I have detailed greatly additional features that may be added in the future.

Tellermate – LiveDrawer

“The intelligent cash drawer builds Tellermate count-by-weight technology into the drawer itself, giving you full visibility of exactly how much money is in the drawer at any one time.” – Tellermate Product Launch 2019 (17)



(18)

Figure 9 LiveDrawer

This piece of hardware is relatively new to the market, it works by weighing notes and coins individually in their compartments, converting this to a till running total. By linking this to the EPOS software in use, you are able to tell when a till has excessive notes or needs change.

In the future work section of this report, I will explain how this piece of hardware can be linked into my software, eliminating cash counting, reducing cash loss and saving significant hours labour.

2.5 Resultant Findings and Requirements

Requirements Analysis

Software requirements can be functional or non-functional. A functional requirement means providing a specific service/action to the user. A non-functional requirement can be user response time or page refresh time, this does not affect the operation of the system but can provide a better user experience. Most of the time requirements are decided during BA analysis, but they can be split into the following categories:

- Business Requirements – These are high level requirements that the business generate
- Architectural and Design Requirements – These are a lot more detailed than business requirements, they really dig down to design level
- System and Integration Requirements - These are the lowest level of requirements and they go into the most detail. These requirements are usually where a developer starts because they are in abundance. (19)

Resultant Findings

After much research and analysing it, I found out the exact needs of a retail EPOS system for the small retailer. A lot of solutions out there have missing non-mandatory features, but implementing these features would be extremely useful for staff and would cut down transaction time dramatically.

Drawing on my near 5-year experience in the retail industry, I understand the everyday frustrations retailers face with their software. As part of my requirements analysis, I am developing new and improved features, some of these ideas are my own, others have come from my co-workers and their experience at both management and shop floor level.

An application for retail needs to be relatively simple to use, elegant in design and have quite a large amount of error prevention. A large variety of people work in the retail industry widely ranging in age and experience, so the screen should be self-explanatory and easy to follow.

I made an online survey to find out what kind of features retailers currently have and what kind of features their employees would like to see. The survey was online for 8 weeks and some of the results are outlined. These ideas helped me optimise my system, to benefit the small retailer.

<https://www.surveymonkey.com/r/CN19DIT>

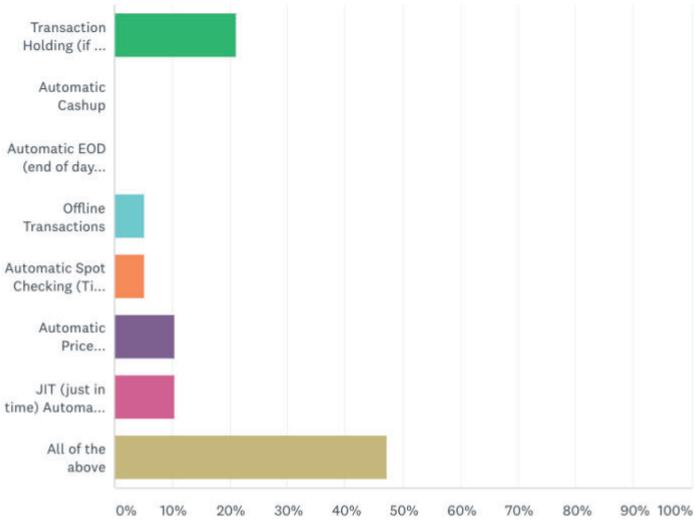
Q5

Customize

Save As ▾

What additional features would you like to see in your systems ?

Answered: 19 Skipped: 1



ANSWER CHOICES	RESPONSES
Transaction Holding (if a void is done, staff do not need to re-scan items)	21.05% 4
Automatic Cashup	0.00% 0
Automatic EOD (end of day figures, backup)	0.00% 0
Offline Transactions	5.26% 1
Automatic Spot Checking (Till counts as money goes in and out)	5.26% 1
Automatic Price Promotions/ Special Offers	10.53% 2
JIT (just in time) Automatic Stock Ordering	10.53% 2
All of the above	47.37% 9
TOTAL	19

Comments (0)

From the analysis, research and findings, explicit user requirements can be created with abundant detail.

3. Design

3.1 Overview

A structured development, test and project plan are vital from the get go of large software development. There are a number of different potentially applicable software development methodologies, for this project.

3.2 Approach & Methodology

One of these such methodologies is feature driven development (FDD), it mixes best practices to deliver what is important to the client. Meaning development is focused on the features that the client values and the functions they expect. (20)

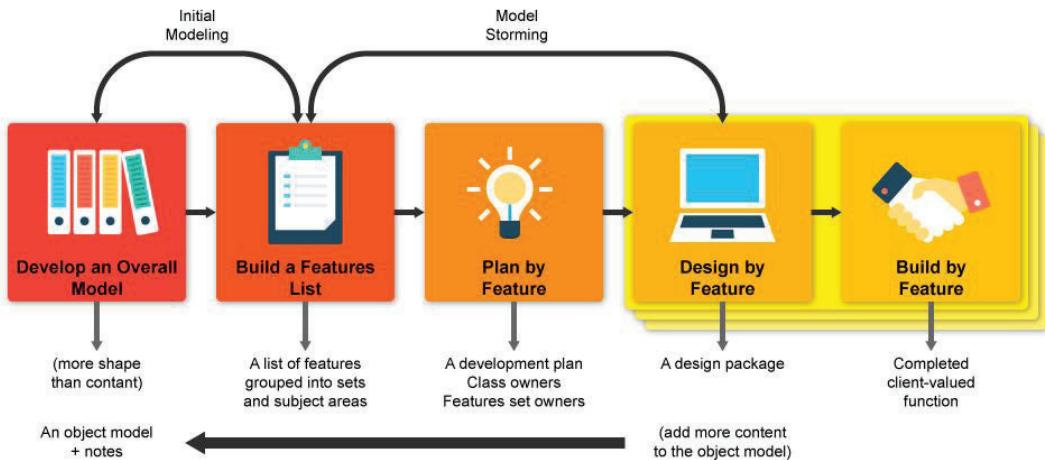


Figure 10 FDD Example

The longest methodology is Waterfall, although I am not using it in this project I think it is very important to mention. It's a linear approach to software development, each stage has a specific purpose and generally one stage must finish before the next one can begin. The full scope of the project is agreed with the customer before development begins. (21)



Figure 11 Waterfall Model

Another very common methodology is Agile Scrum, it is currently used widely across industry from government to the healthcare sector. The key components include: user stories, sprints, sprint backlog, daily stand ups (15 minute meetings), incremental/iterative development and milestone retrospectives (sprint reviews). Sprints are the very core of this methodology and usually last anywhere from 2-4 weeks. Before each sprint starts there is a sprint planning session to prioritise important features to develop for that sprint. (22)



Figure 12 Agile Scrum

This project is taking a hybrid approach and using both Feature Driven Development and Agile Scrum for planning. Both of these methodologies are normally used as part of a team, although in the past they have been used for solo projects. For this project, they are being adapted for use by a one-man team.

3.3 Technical Diagrams

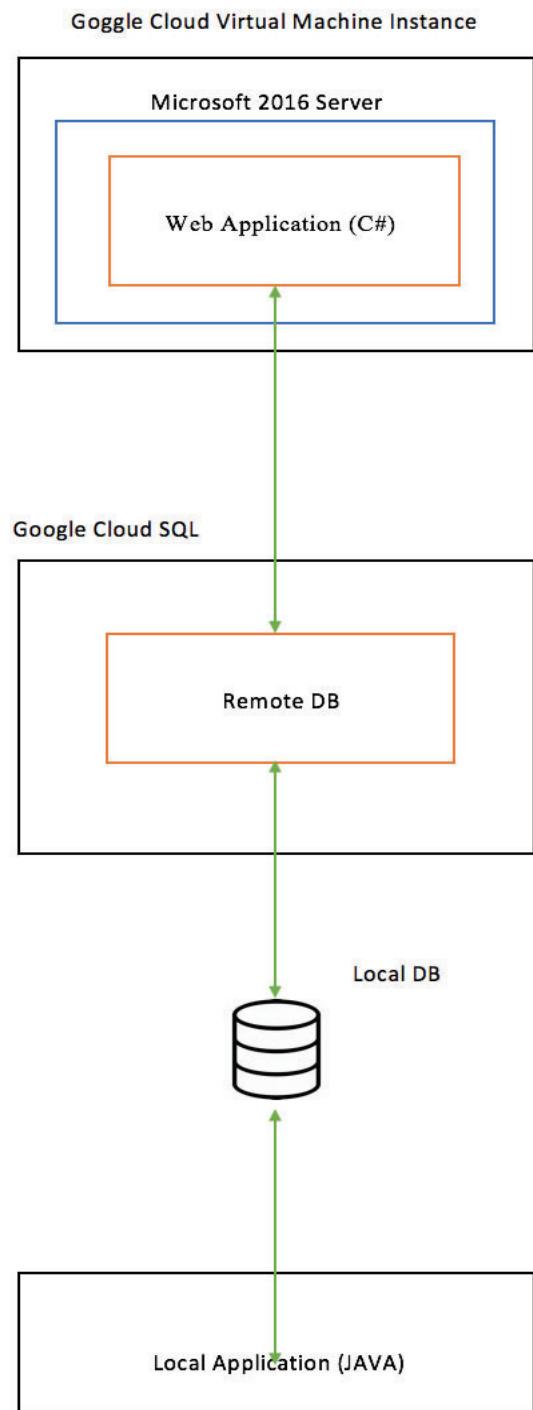


Figure 13 Architecture Diagram

The previous Architecture Diagram shows a high-level representation of how the local application and the remote web application interact and store data. Two different databases are used, one to store data locally, which syncs with the remote to ensure no data is lost. The other is the remote database on google cloud, this ensures data is always up to date and safe.

Entity Relationship Diagram:

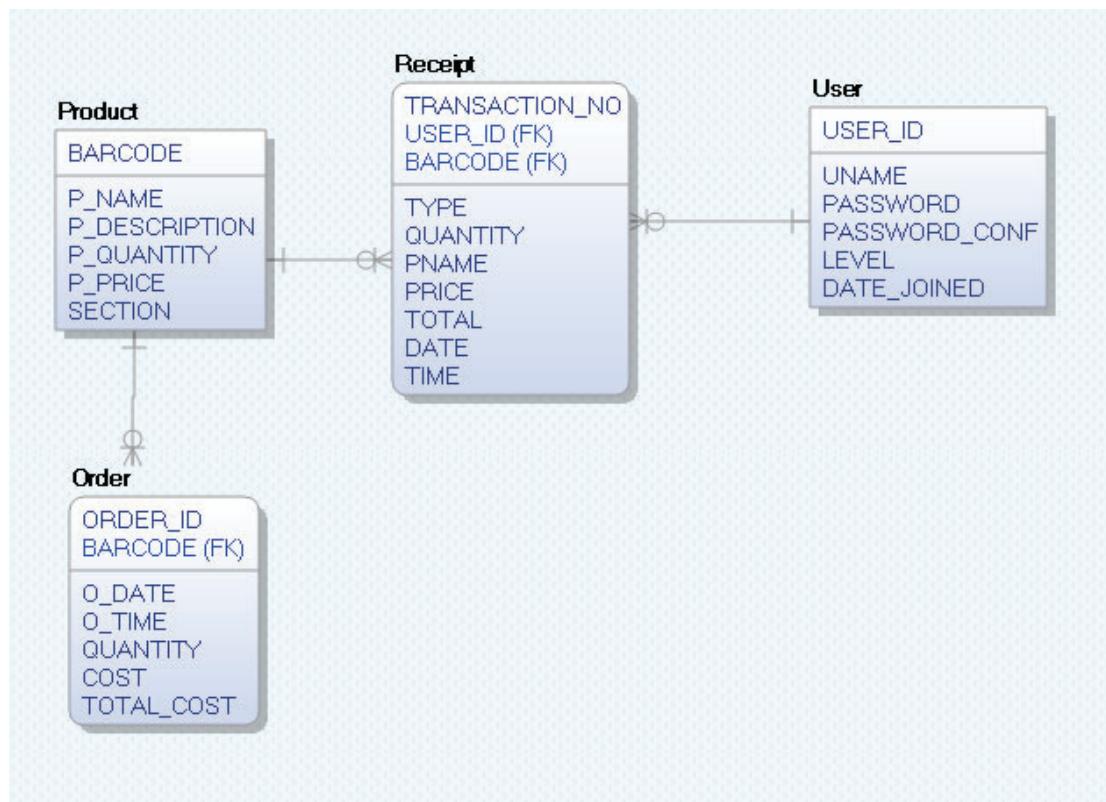


Figure 14 ERD Diagram

The diagram above represents the MySQL databases that these applications are going to use, local and remote (cloud) databases are the same. Outlined below are the table definitions:

Receipt

The receipt table stores all the information to do with a particular sale. Once a sale is completed all the information relating to that sale is recorded in this table. The primary key TRANSACTION_NO uniquely identifies a transaction, the USER_ID is also recorded for tracking purposes.

- TRANSACTION_NO: The primary key, acts as a unique identifier
- TYPE: The transaction type i.e. SALE
- BARCODE: The code that identifies a product
- QUANTITY: The number of products being purchased (i.e. x2, x5)
- PNAME: The name of the product
- PRICE: The selling price of the product
- TOTAL: The total amount for that receipt
- DATE: The date the product(s) were purchased
- TIME: The time the product(s) were purchased
- USER_ID: The person who made the sale

User

The user table contains all the information about the employee (user) who is currently signed on. The table is inserted into when a new employee is added, but it is used constantly for signing in/out of the system. The primary key is USER_ID which is unique.

- USER_ID: Essentially the employees number
- UNAME: The name the employee logs on with
- PASSWORD: The selected password for that particular user
- PASSWORD_CONF: To confirm the password matches during registration
- LEVEL: The seniority of the user, where 1: Manager, 2: Supervisor, 3: Cashier, Cashiers have basic functions while Manager/Supervisors have extra functions
- DATE_JOINED: The date the employee was registered

Product

The product table contains the information relating to all products. The table is inserted into when an order is received, but it is also deducted from when a product is sold. The primary key is the BARCODE of the product, due to EAN/UPC the barcode is already unique to a particular product.

- BARCODE: The universal product code that identifies a product
- P_NAME: The name of the product
- P_DESCRIPTION: Short description of the product
- P_QUANTITY: The amount of a particular product in stock (i.e. milk 12)
- PRICE: The selling price of the product
- SECTION: The sub section the product belongs to (i.e. chilled products, canned foods, confectionary etc)

Order

The order table contains information for replenishing products. The table is used to order new stock when product levels fall below ten units. The primary key is the ORDER_ID which uniquely identifies that specific order.

- ORDER_ID: The number that identifies an order
- O_DATE: The date the order was sent
- O_TIME: The time the order was sent
- BARCODE: The product barcode that is being ordered
- QUANTITY: The quantity being ordered (i.e. milk, 22)
- COST: The cost price of a product
- TOTAL_COST: The total cost of the order

Use Case Diagram:

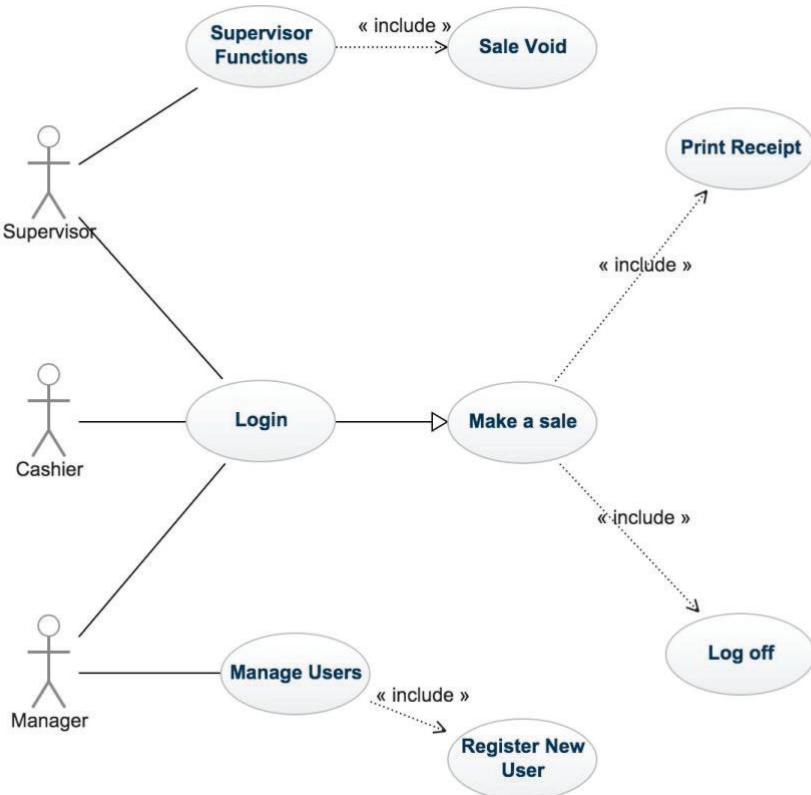


Figure 15 Use Case Diagram

The above Use Case Diagram explains how the user(s) with different permissions, will interact with the system. The above use cases are the only features the user is able to interact with, they will not have access to any of the core components or inner workings of the system.

- Cashier: Has the ability to log in, make a sale, log off
- Supervisor: Has the ability to log in, make a sale, access supervisor functions such as void sale, log off
- Manager: Has the ability to access all the features of the system including the manage users function

Sequence Diagram:

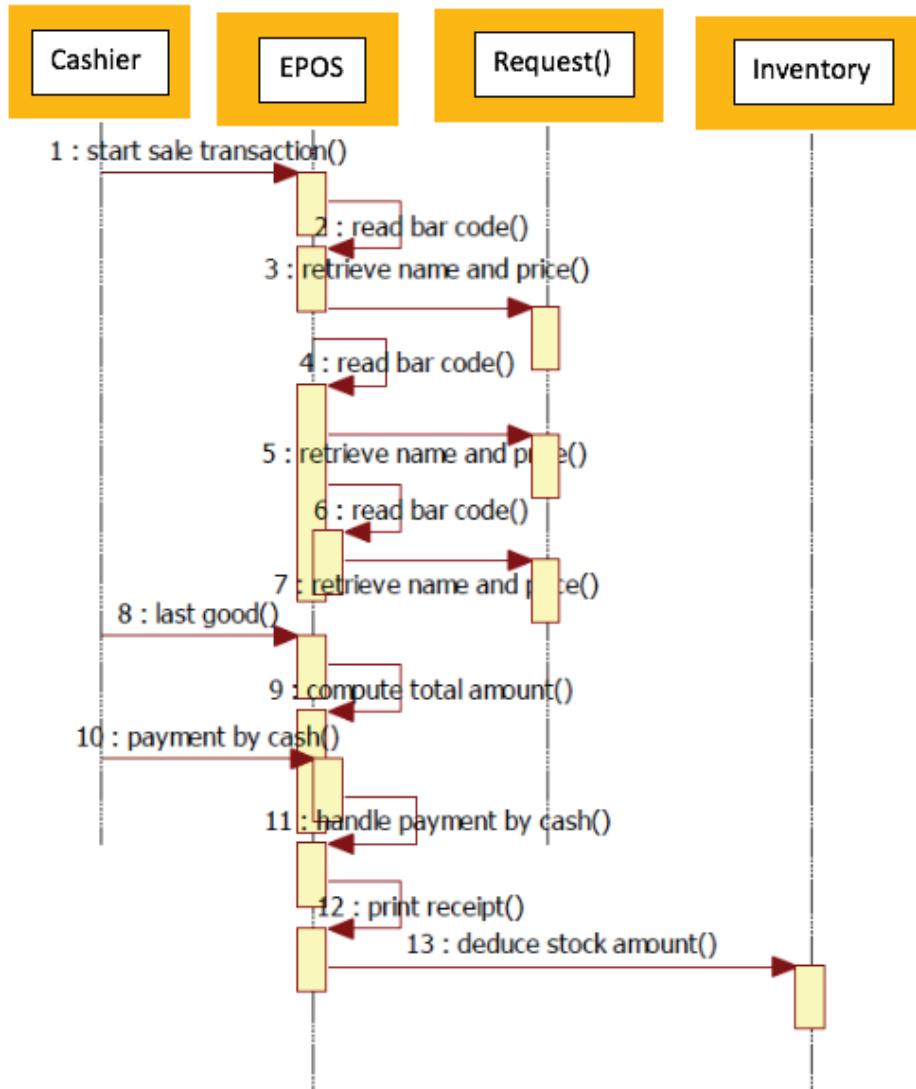
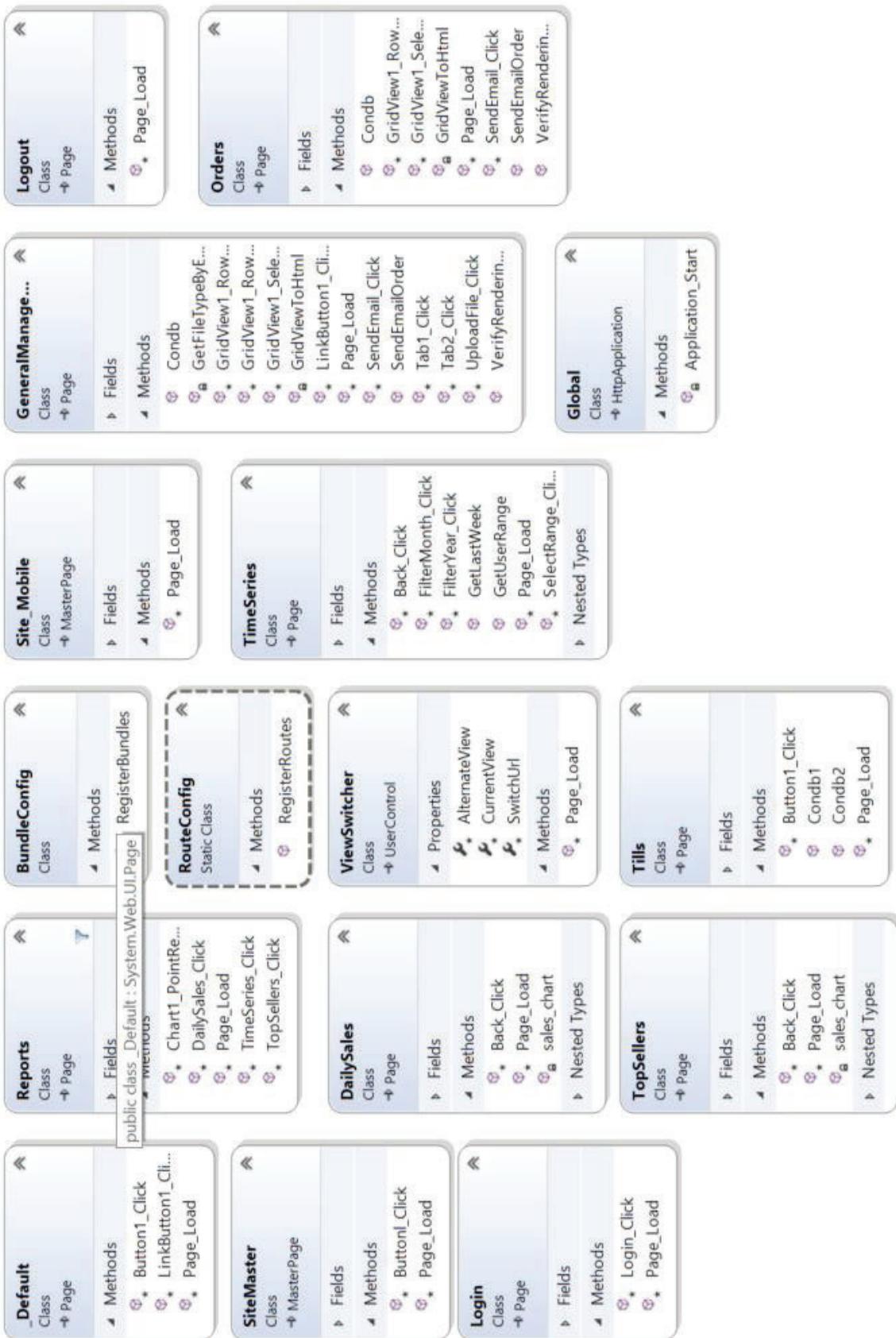


Figure 16 Sequence Diagram

The sequence diagram shows the main methods a cashier interacts with when performing a transaction. It also shows how the EPOS system interacts with the inventory to deduct stock as its scanned.

The simplified class diagram below shows the main classes used by the RetailOnline web application. It does not show all classes as the diagram would be too large. It does show the main classes and methods.

Class Diagram:



3.4 User Experience & Interface

“User experience (UX) is the overall experience of a person using a product such as a website or computer application, especially in terms of how easy or pleasing it is to use” Outlined below are the main components of ux: (23)



Figure 18 Peter Morville’s UX Honeycomb

While user experience and user interface are often lumped into the same category, they are actually two completely different areas. User interface (UI) is focused on how the application / system looks from a design perspective, user experience (UX), takes that design and make sure it resonates with user based on the above diagram.

There is a great analogy out there that explains the difference between the two. Imagine a cruise ship, the internal workings are the code, the UX is everything inside that ship that makes it move, while the UI is everything on the outside that makes the ship look nice and appealing.

4. Implementation and Development

This section covers the development of the system. The system is based across 3 levels, the application, the web server and the middle layer (data access layer + security layer).

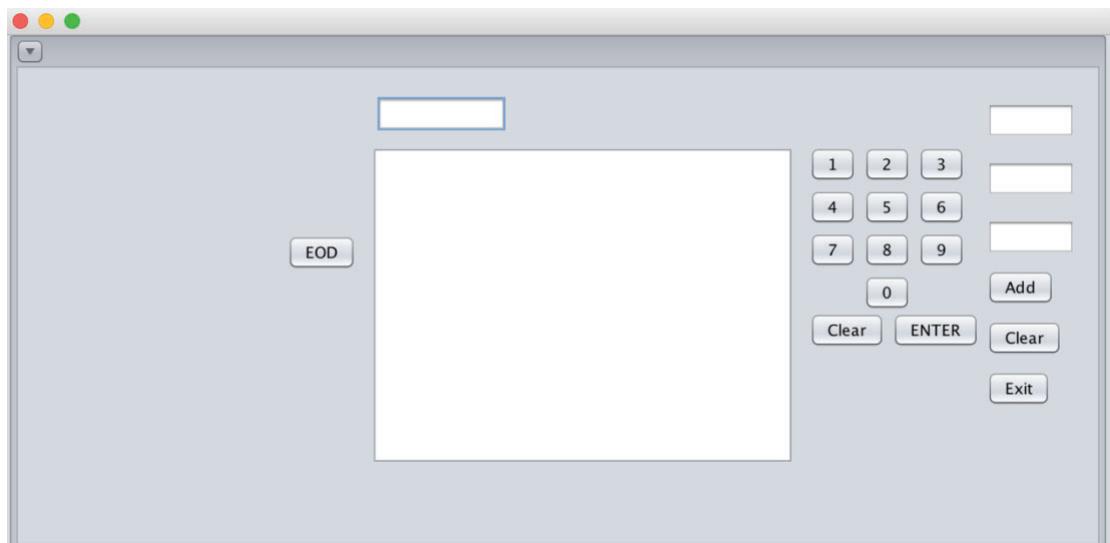
4.1 Detailed System Overview

Local Application

The local application that runs on the clients computer, it was developed using Java through the NetBeans environment. All code is class based and jFrames are used to implement components in the GUI. The diagrams below shows the system:

Prototype :





Development Finished:

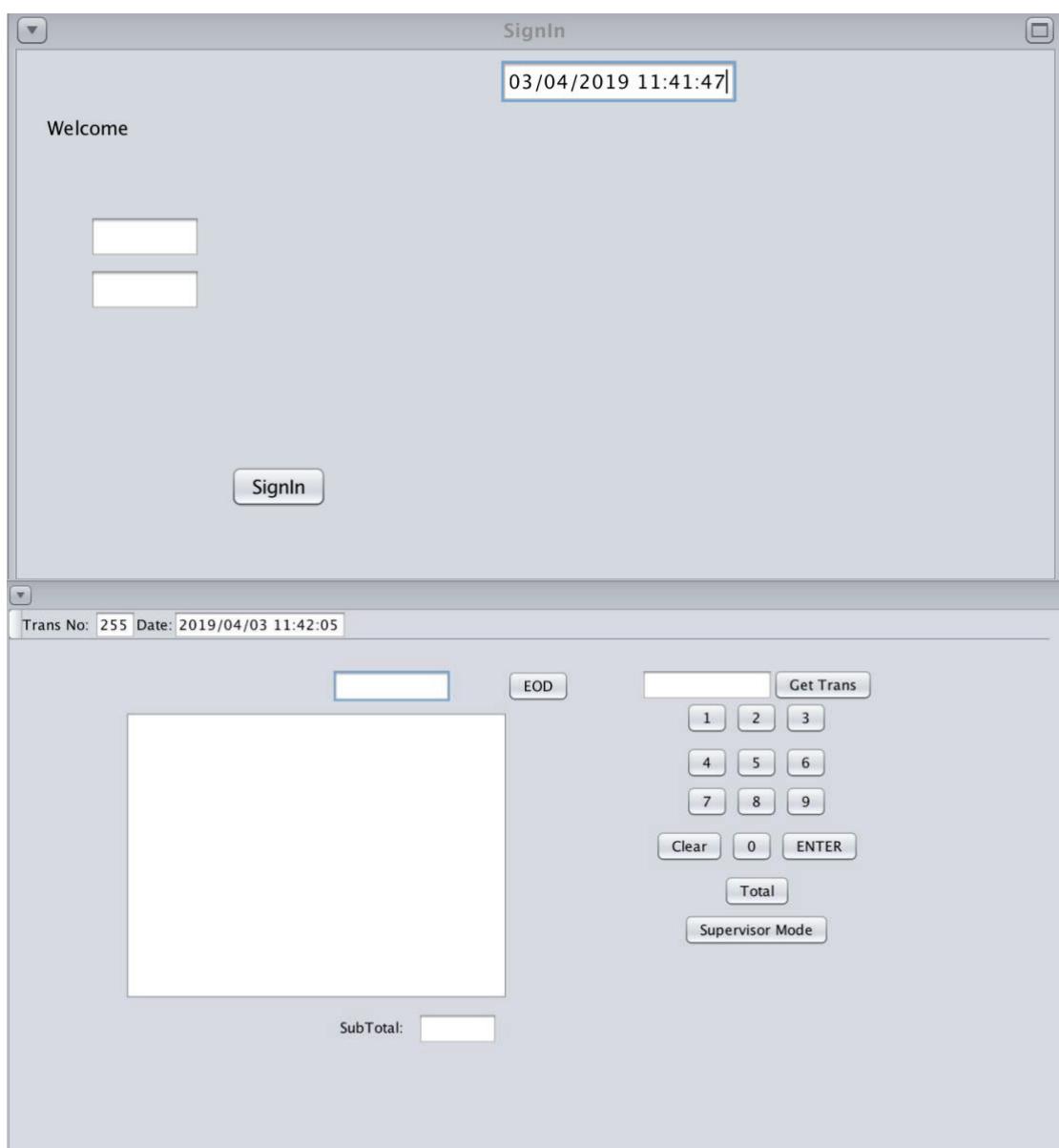




Figure 19 Local Application

Java was chosen as the language for the local application because it's completely independent across all platforms. So, it does not matter what operating system the client's machine is running, the application will run in almost any environment. As per the screenshots above I was running this on Mac.

The below diagram shows the java classes and files used in this project. Including the Java Database Connector (JDBC), which is a jar file in the bin folder.

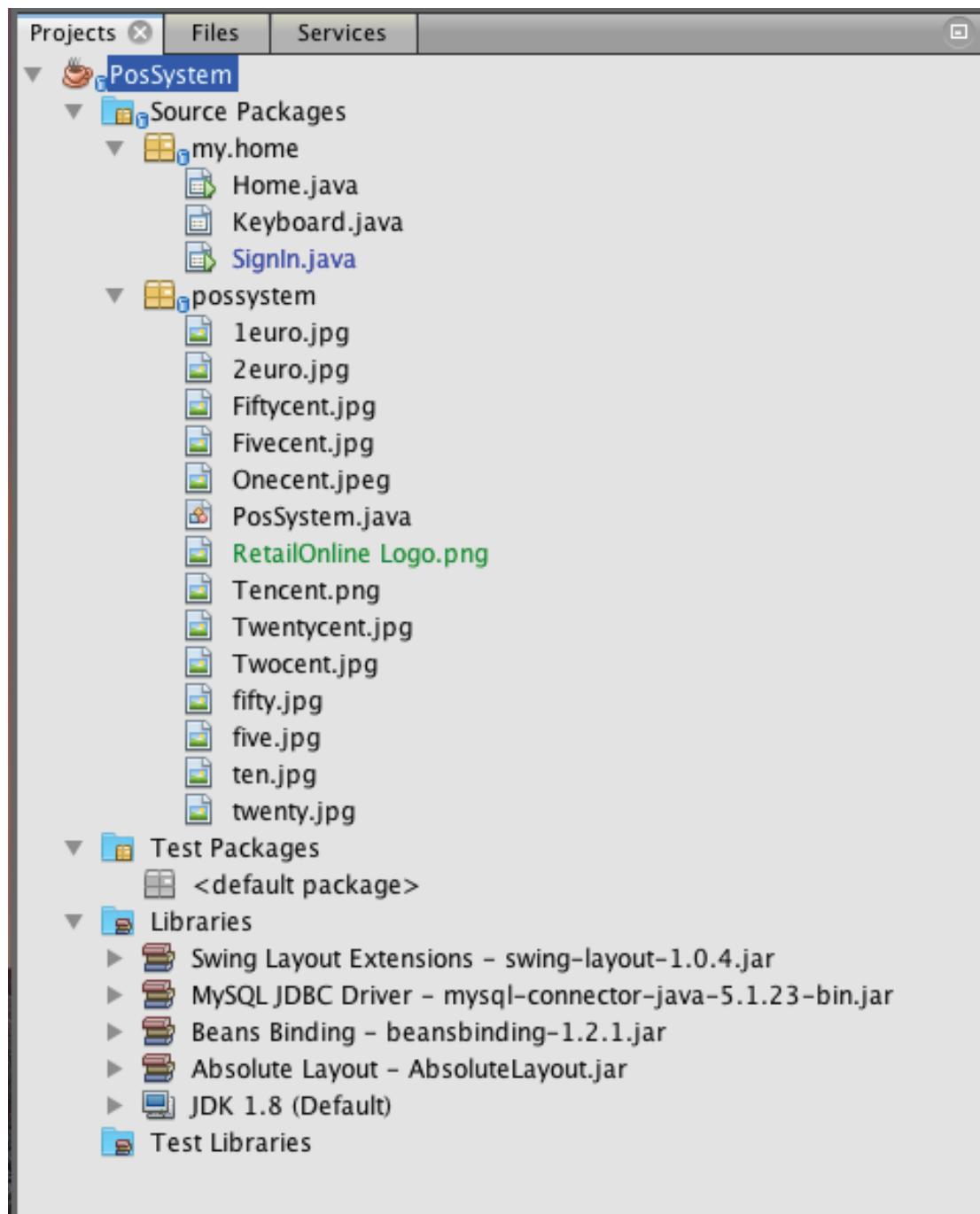


Figure 20 NetBeans Project Layout

Web Application

The web app runs on a google cloud virtual machine instance, on this vm microsoft server 2016 is installed. All code is pushed onto this server, I bought a domain name called “possolutions.ie” which is linked to the IP address of the server. The webapp can be accessed from anywhere through this web address but the user must have valid credentials to log into the system.

Prototype:

The screenshot shows a browser window with the title "Log in - My ASP.NET Application". The URL in the address bar is "possolutions.ie/Account/Login". The page has a dark header with "Application name", "Home", "About", "Contact", "Test", "Register", and "Log in" links. The main content area has a heading "Log in." and a sub-instruction "Use a local account to log in.". It contains two input fields: "Email" and "Password". Below these is a "Remember me?" checkbox and a "Log in" button. To the right, there is a note about external authentication services and a link to an article. At the bottom, there is a "Register as a new user" link and a copyright notice: "© 2018 - Craigs Web Technologies".

Development Finished:

The screenshot shows a browser window with a dark header containing the "RetailOnline" logo and navigation links for "Home", "Reports", "Till Management", and "General Management". The main content area has two input fields (Email and Password) and a "Login" button. At the bottom, there is a copyright notice: "© 2019 - RetailOnline2019" and a "Logout" button.

Reports page:

RetailOnline Home Reports Till Management General Management

Top Sellers Daily Sales By Operator Time Series

© 2019 - RetailOnline2019

Logout

Daily Transactions

A bar chart titled "Daily Transactions" comparing the number of transactions for two operators. The Y-axis represents the "Number of Transactions" from 0 to 55. The X-axis represents the "Operator Number". Operator 1234567 has 50 transactions, and Operator 12345678 has 18 transactions. A tooltip indicates the value for Operator 1234567.

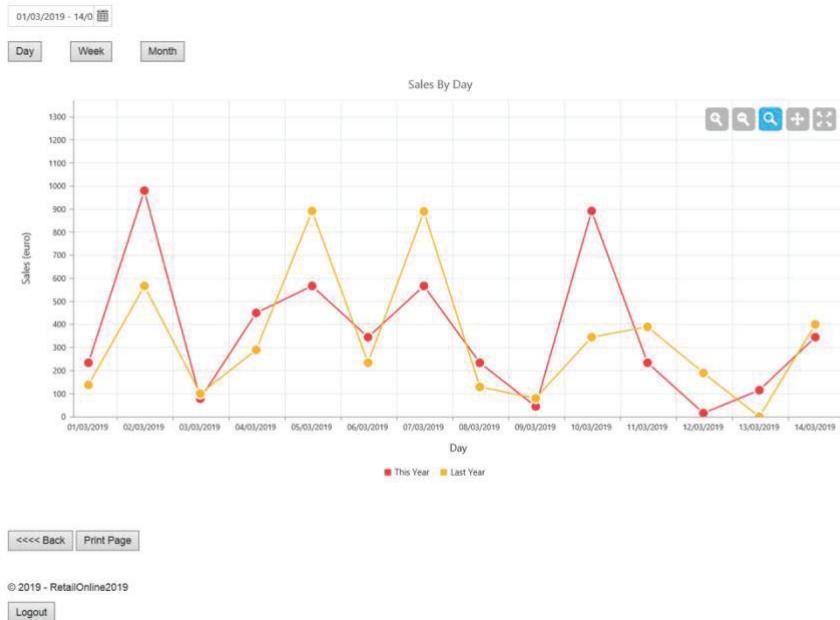
Operator Number	Transactions
1234567	50
12345678	18

Number of Transactions

Operator Number

Transactions

<<< Back Print Page



Till Management Page:

EPOS - Till 001

€50	€20	€10	€5		
3650	660	100	235		
€1	€2	50c	20c	10c	5c
-11	28	2.50	-6.00	7.03	0.50

© 2019 - RetailOnline2019

General Management Page:

File Server Orders Tab 3

Upload Files:

File	File Size (bytes)	File Type
hello.xls	157184	Microsoft Excel Document
System.Web.UI.WebControls.TextBox	42496	Unknown
test	157184	Unknown
test.xls	157184	Microsoft Excel Document
test1	157184	Unknown
test2	157479	Unknown
test3.xls	42496	Microsoft Excel Document
test45	157184	Unknown
test45.xls	157184	Microsoft Excel Document
testsf.xls	157184	Microsoft Excel Document
tioejstd.xls	157184	Microsoft Excel Document

© 2019 - RetailOnline2019

File Server **Orders** Tab 3

Barcode	Name	Description	Quantity
1122334455	milk	dairy	<input type="text"/>
1122334456	chewing gum	chewy	<input type="text"/>
1122334457	bread	bakery	<input type="text"/>
5011041001180	white bread	bakery	<input type="text"/>
5011044001057	butter	chilled goods	<input type="text"/>
5099274000118	popcorn	conf/sweets	<input type="text"/>
509839301803	salt	long lasting	<input type="text"/>
5397136002998	brown sauce	long lasting	<input type="text"/>

© 2019 - RetailOnline2019

Figure 21 Web Application possolutions.ie

C sharp was chosen as the language for the server side. It provides more functions for handling data analysis and formatting reports. It works well with the .NET framework that I am using as it's a Microsoft language. The web app is constantly available from any location because it runs in the cloud. Meaning that it can be accessed from any device with any operating system, this even includes mobile devices. The below diagram shows, the cshtml (C# + html) files, the hierarchy of the web app and all other source files including configuration files.

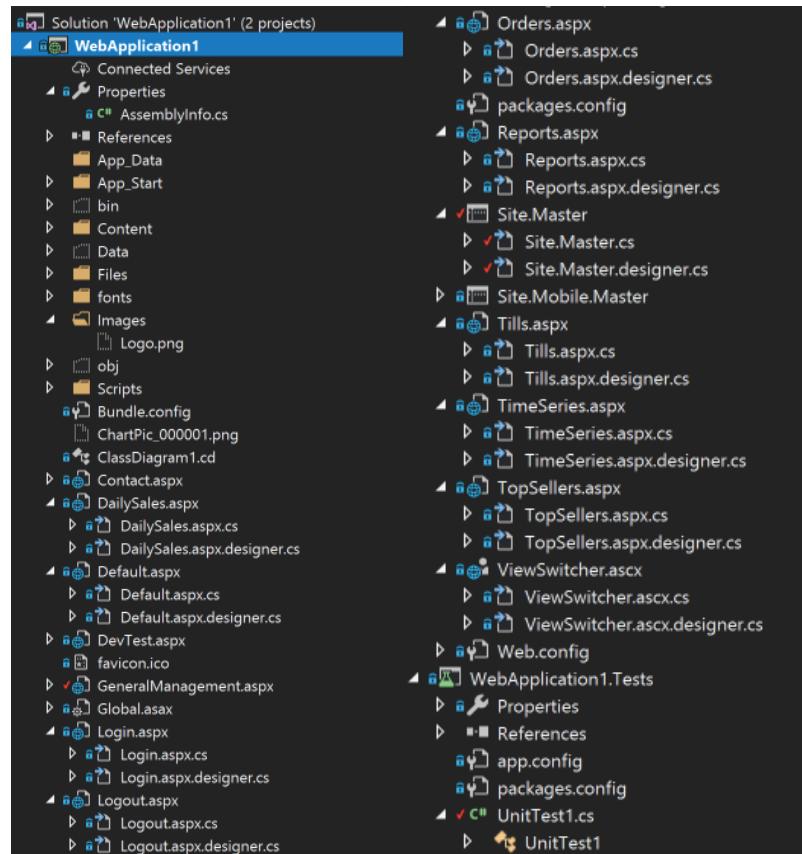


Figure 22 Visual Studio Project Layout

Cloud Setup & Linking IP

VM instances

aspnet-3

Zone: europe-west3-a | Recommendation: Increase perf.

Internal IP: 10.156.0.3 (nic0) | External IP: 35.242.254.68 | Connect: RDP

Figure 23 Cloud Server

This is the Virtual Machine that is running the Microsoft server 2016 edition, it also contains the code for the web application. To access the server, I am using a remote desktop connection along with my credentials and the external IP of the VM, as the administrator I am able to make changes to the server.

The domain name for the application was bought on Register365.com, I then edited the DNS settings, so that possolutions.ie points to the external IP address of the Google VM. Essentially this points the domain name to the server, as per the detailed image below:

A, CNAME, AAAA, TXT and NS records			
#	Host name	Type	Result
1	Google VM	A	35.242.254.68
2	imap	CNAME	imap.reg365.net
3	pop3	CNAME	pop3.reg365.net
4	www	CNAME	fwd3.hosts.co.uk
5		TXT	google-site-verification:...
6		TXT	v=spf1 include:spf.reg3...
7		A	
8		A	

Mail exchange records			
#	Host name	Priority	Result
1		30	mx1.reg365.net
2		30	mx2.reg365.net

Service records					
#	Name	Priority	Weight	Port	Result
1	_autodis		0	443	autodiscover.reg365.net
2					

CANCEL **DELETE** **SAVE**

Figure 24 DNS Settings

Cloud SQL

The remote database runs on an instance in cloud SQL. The web application connects to this instance through the website's IP address. In other words only connections through possolutions.ie will be accepted for selecting and inserting into the database. All other connections will be refused for security reasons, at the moment I have a second IP address allowed, which is the IP address of my computer. This is purely just for development and testing purposes. The cloud database is shown in the screenshot below, like the local database the cloud one is also MySQL.

The screenshot shows two views of a Cloud SQL instance named 'instance-pos'. The top view is the 'Databases' tab, showing a list of MySQL databases:

Name	Character set	Collation	Type
information_schema	utf8	utf8_general_ci	System
mysql	utf8	utf8_general_ci	System
performance_schema	utf8	utf8_general_ci	System
pos	utf8	utf8_general_ci	User

The bottom view is the 'IP ADDRESS' tab, showing the IPv6 and IPv4 addresses assigned to the instance:

IPv6 address
Every First Generation Cloud SQL instance has an IPv6 address free of charge.
1a78:32f1:2d8a

IPv4 address
You will be charged \$0.01 each hour the instance is inactive and has an IPv4 address assigned.
.224.26

Release IPv4 address

Figure 25 Cloud SQL

GitHub

In this project git is being used for version control and as a code repository, I created a repository called Final-Year-Project. There are two different branches in this repository, one called master, the other called Local. Master contains all the files for the webserver and the web application, while local contains all the files for the local EPOS system.

The screenshot shows the GitHub repository page for 'craig0222 / Final-Year-Project'. The 'Code' tab is selected. At the top, there are statistics: 27 commits, 3 branches, 0 releases, and 1 contributor. Below this, a list of commits is shown:

File	Commit Message	Date
.vs	logo and colour added	3 days ago
WebApplication1.Tests	timeSeries parsing now fixed	10 days ago
WebApplication1	logo and colour added	3 days ago
packages	Time series now added, for sales reports	2 months ago
CreateTables.sql	creates.sql backup file created for DB	8 days ago
README.md	Read me added back deleted by mistake	2 months ago
WebApplication1.sln	reports now working	2 months ago

Figure 26 Github master branch

The screenshot shows the GitHub repository page for 'craig0222 / Final-Year-Project'. The 'Code' tab is selected. At the top, there are statistics: 26 commits, 3 branches, 0 releases, and 1 contributor. A note says 'This branch is 26 commits ahead, 27 commits behind master.' Below this, a list of commits is shown:

File	Commit Message	Date
.vs	Both Databases Synced	5 months ago
PosSystem	UI Design & Logo	3 days ago
RetailSystem	Both Databases Synced	5 months ago
packages	Initial project setup	6 months ago
.DS_Store	Predictive Change Now Implemented	23 days ago
.gitignore	Final Local System Submission for Interim	4 months ago
README.md	Updated readme	4 months ago
RetailSystem.sln	initial project setup	6 months ago

Figure 27 Github local branch

Trello – Online Kanban Board

Throughout this project Trello Kanban Board was used every day to add new features and update existing ones until they were development complete and ready for system testing.

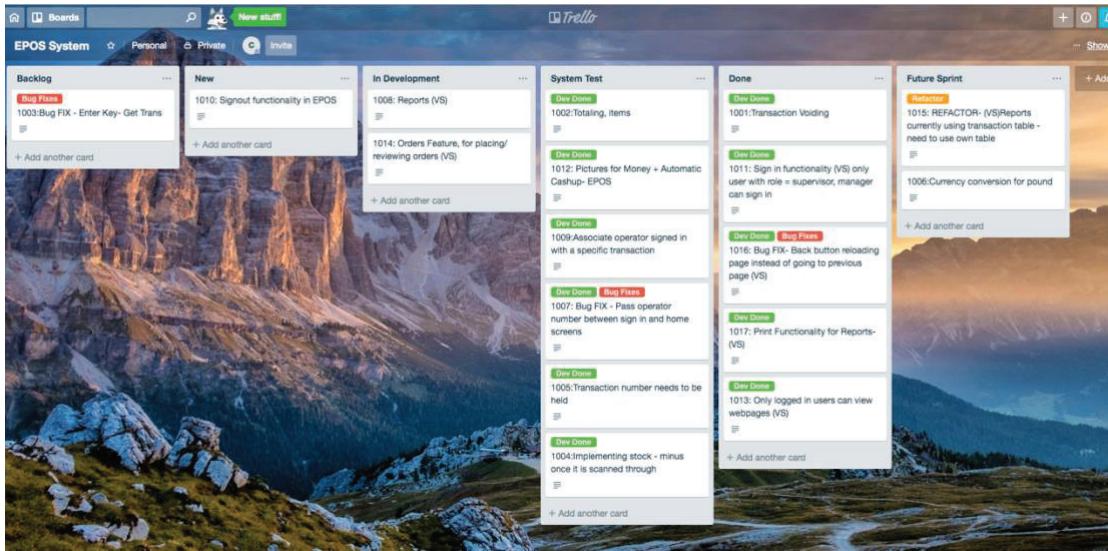


Figure 28 Trello Kanban Board

Each ticket above is assigned a unique number which identifies that ticket, for this project as there is only one developer it is not extremely important because I am the one inputting the tickets. If you were part of a big development team, as on my placement, it would be necessary to number tickets and place a priority on them this is normally done by the Business Analyst (BA).

The tickets would then be inputted to a sprint management system such as Microsoft's Team Foundation Server (TFS). From there, the scrum master or senior development manager would assign the tickets to different members of the development team.

For this project I have added tags, this allows me to know which tickets are new features, which ones are bug fixes and other ones that are refactors. A refactor is done to make client-side code execution faster, to clean up poorly written code, to comply with GDPR and to improve overall performance.

The use of trello throughout this project has saved a great deal of time. It keeps tickets organised and in the one place, easily accessible at any given time.

4.1 Development Process

Money Counting & Exact Change

Explained below is the algorithm I came up with, to let the cashier know exactly what notes and coins to give the customer based on the money the money the cashier received from the customer.

```
1 if (changeToBeGiven > 0 && TotalBtn.pressed() == true)
2 {
3     int cents = (int)(changetotal*100);
4
5     while (change >= fiftyEuro && fiftyEuroNote > 0)
6     {
7         int fiftyEuro = tenEuro/5;
8         tenEuro = tenEuro%5;
9     }
10    while (change >= twentyEuro && twentyEuroNote > 3)
11    {
12        int twentyEuro = tenEuro/2;
13        tenEuro = tenEuro%2;
14    }
15    while (change >= tenEuro && tenEuroNote >= 10)
16    {
17        int tenEuro = fiveEuro/2;
18        fiveEuro = fiveEuro%2;
19    }
20    while (change >= fiveEuro && fiveEuroNote >= 15)
21    {
22        int fiveEuro = oneEuro/5;
23        oneEuro = oneEuro%5;
24    }
25    while (change >= twoEuro && twoEuroNote >= 15)
26    {
27        int twoEuro = twoEuro/200;
28        twoEuro = twoEuro%200;
29    }
30    while (change >= oneEuro && oneEuroNote >= 10)
31    {
32        int oneEuro = cents/100;
33        cents = cents%100;
34    }
35    while (change >= fiftyCent && fiftyCentCoin >= 10)
36    {
37        int fiftyCent = cents/50;
38        cents = cents%50;
39    }
40    while (change >= twentyCent && twentyCentCoin >= 10)
41    {
42        int twentyCent = cents/20;
43        cents = cents%20;
44    }
45    while (change >= tenCent && tenCentCoin >= 10)
46    {
47        int tenCent = cents/10;
48        cents = cents%10;
49    }
50    while (change >= fiveCent && fiveCentCoin >= 10)
51    {
52        int fiveCent = cents/5;
53        cents = cents%5;
54    }
55 }
56 else
57 {
58     print("NO CHANGE TO BE GIVEN");
59 }
```

Figure 29 Simplified Code of Change Algorithm

- It works by consuming the larger notes before giving out smaller change. This way it ensures that the maximum amount of small change in the till stays above a given level. Thus reducing the number of times a day the till will need to be replenished with change.
- The system is also able to calculate the change the cashier needs, therefore reducing the amount of human error. From a corporate point of view it is easy to determine the amount of money in the till at any given time, giving a breakdown of the exact notes and coins. This allows management to replenish tills promptly when they see a particular note/coin running low.

SMTP Server & Automatic Emails

- When a member of management places an order, an email is automatically sent directly to the central warehouse for picking. This means faster ordering and less time spent in the office for a manager. This is done through SMTP.
- The SMTP server (Simple Mail Transfer Protocol) is the email server responsible for sending emails. One SMTP server can transfer the mail to another SMTP server and relay it through several re directs to the destination. It is able to identify the recipient (in this case the warehouse) using the recipients id (retailonlineorders@gmail.com).
- The code snippet below shows the main implementation of this through the orders.aspx page, although other pages use code to interact with this server to get the current managers email address, current date/time of order, products, quantity of items and so on. It is done through the SendEmailOrder() method of the order class.

```

86     public void SendEmailOrder()
87     {
88         SmtpClient sc = new SmtpClient("mail address");
89
90
91         try
92         {
93             MailMessage mailMessage = new MailMessage();
94             mailMessage.To.Add("retailonlineorders@gmail.com");
95             mailMessage.From = new MailAddress("retailonlineorders@gmail.com");
96             mailMessage.Subject = "Order";
97             mailMessage.Body = GridViewToHtml(GridView1);
98             mailMessage.IsBodyHtml = true;
99
100            SmtpClient smtpClient = new SmtpClient("smtp.gmail.com", 587);
101            smtpClient.Credentials = new System.Net.NetworkCredential()
102            {
103                UserName = [REDACTED]
104                Password = [REDACTED]
105            };
106            smtpClient.EnableSsl = true;
107            smtpClient.Send(mailMessage);
108            //Response.Write("E-mail sent!");
109
110        }
111        catch (Exception ex)
112        {
113            Response.Write("Could not send the e-mail - error: " + ex.Message);
114        }
115    }
116

```

Figure 30 SMTP Server Code Snippet

Cloud File Storage

This feature was not included in the original scope, but as research was ongoing I found out most small retailers save all their files (such as staff rosters, budgets, promotion signs etc, which are all normally either excel, onenote or word files) locally on their pc or on an instore server.

I came up with the idea of having an option for file storage on the remote (web) application. All the user has to do is upload the file once and its attached to their account until they delete it. It works by user token, so when a certain user uploads a file the system only gives them access to that file. In the future I hope to take this feature further to include file sharing with other users in the system.

The screenshot shows a user interface for file storage. At the top, there is a 'Browse...' button for selecting files. Below it is a 'Button' control. A table displays a list of uploaded files with columns for 'File', 'File Size (bytes)', and 'File Type'. The data in the table is as follows:

File	File Size (bytes)	File Type
hello.xls	157184	Microsoft Excel Document
System.Web.UI.WebControls.TextBox	42496	Unknown
test	157184	Unknown
test.xls	80384	Microsoft Excel Document
test1	157184	Unknown
test2	157479	Unknown
test3.xls	42496	Microsoft Excel Document
test45	157184	Unknown
test45.xls	157184	Microsoft Excel Document
testtsf.xls	157184	Microsoft Excel Document
tioejjsfd.xls	157184	Microsoft Excel Document

Figure 31 File Server Upload Feature

```
protected void UploadFile_Click(object sender, EventArgs e)
{
    //this method is used for file uploading
    String docName;
    docName = TextBox1.Text.ToString();
    if(DocUpload.HasFile)
    {
        //string fileName = DocUpload.FileName;
        DocUpload.PostedFile.SaveAs(Server.MapPath("173.194.225.114/GCS/Bucket") + docName);
    }

    DataTable dt = new DataTable();
    dt.Columns.Add("File",typeof(string));
    dt.Columns.Add("Size",typeof(string));
    dt.Columns.Add("Type",typeof(string));

    foreach (string strfile in Directory.GetFiles(Server.MapPath("173.194.225.114/GCS/Bucket")))
    {
        FileInfo fi = new FileInfo(strfile);
        dt.Rows.Add(fi.Name, fi.Length,
                    GetFileTypeByExtension(fi.Extension));
    }

    GridView1.DataSource = dt;
    GridView1.DataBind();
}

protected void GridViewFile_RowCommand(object sender, GridViewCommandEventArgs e)
{
    //this method is used to download a file
    Response.Clear();
    Response.ContentType = "application/octet-stream";
    Response.AppendHeader("Content-Disposition", "filename=" +
                           e.CommandArgument);
    Response.TransmitFile(Server.MapPath("~/Files/")
                           + e.CommandArgument);
    Response.End();
}
```

Figure 32 Connecting to File Server Code Snippet

Checking the extension of the uploaded document, is not essential but it is beneficial to do as then I am able to tell the user the types of documents they have uploaded. They could have two documents that are two different file types with the same name, so knowing which document is excel for example can save the user time.

```
private string GetFileTypeByExtension(string fileExtension)
{
    //used to get the extension of the given file
    switch (fileExtension.ToLower())
    {
        case ".docx":
        case ".doc":
            return "Microsoft Word Document";
        case ".xlsx":
        case ".xls":
            return "Microsoft Excel Document";
        case ".txt":
            return "Text Document";
        case ".jpg":
        case ".png":
            return "Image";
        default:
            return "Unknown";
    }
}
```

File Type
Microsoft Excel Document
Unknown
Unknown
Microsoft Excel Document
Unknown
Unknown
Microsoft Excel Document
Unknown
Microsoft Excel Document
Microsoft Excel Document
Microsoft Excel Document

Figure 33 Detecting File Extension

Now knowing the file extension, I am able to output to the gridview what the file type is off all files, stored on the server for this current user.

5. System Validation

5.1 Manual Testing

When a new system is created, it needs to be tested thoroughly to catch bugs and problems missed during the development stage. This includes testing the frontend, backend, data access layer, scripts and any configurations the system uses. A lot of the testing so far has been carried out while the system was being developed. As well as testing analysis/evaluation needs to be carried out on the system to ensure it meets user requirements and is easy to use.

5.1.1 Functional

“Functional testing is a software testing process used within software development in which software is tested to ensure that it conforms with all requirements. Functional testing is a way of checking software to ensure that it has all the required functionality that's specified within its functional requirements.” (24)

Smoke Testing

Smoke testing is a high-level check of all features after a build. It ensures that all the main features are still working and the latest code in the build hasn't affected the functionality of these features. I am using smoke testing throughout this project after builds to make sure features are still working.

Regression Testing

Like smoke testing, regression testing tests that all functional & non-functional features are still working. The key difference is smoke testing is performed more often, where regression is used coming up to a release. Currently, I am not using regression testing for the prototype but before the final project submission I will be using it to thoroughly test every aspect of the system. (25)

Interface Testing

Like end to end testing, interface testing tests the communication between an application, server and database by using commands, scripts and messages. (26)

The two main categories of Interface Testing:

1. Web server and application interface
2. Application and Database server interface

The Testing checks that:

- Errors are handled properly or return valid error messages to the user
- Tests the outcomes when connections are reset between server and applications

5.1.2 Non-Functional

“Non-Functional testing is a software testing technique that verifies the attributes of the system such as memory leaks, performance or robustness of the system. Non-Functional testing is performed at all test levels.” (27)

Stress Testing

Is a non-functional test, the idea is to put a large load on the server and see how it performs under “stress”. As my system is for the small retailer, the server will not have to deal with a large number of users at any given time. But it is still crucial to test the performance of the server even with a small number of users.

Security Testing

As data protection is critical these days, security testing is a crucial part of any test plan, testing user’s permissions and roles to make sure they can only access features for their given role. Also, another part of security testing is URL hacking, changing the identifier in the web browsers URL to try gain access to someone else’s account. Then also testing the data access layer to ensure SQL injection is not allowed.

All of the above testing methods were used throughout this project at different points, to ensure good user experience and fast response times.

5.1.3 Test Plan

Scope

The general motivation behind testing is to guarantee that the application meets the majority of its specialized, useful and business prerequisites. The details below describes the overall test plan and the strategies for testing the system.

Test Objectives

The objectives of testing are to ensure complete validation of the software’s requirements:

- Ensure defects are noted
- Correct validation is implemented on all aspects of the application/server
- Ensure only valid data is being stored in the database, triggers and indexes are working correctly

Testing Goals

- Ensuring all business requirements are met
- UX (user experience) is intuitive
- All parts of the system are working correctly

Scenarios Tested

Features Local Application:	Working:	Comment:
Sign In	Yes	
Make A Sale	Yes	Very small bug, cent rounding
Price Check	Yes	
End of Day (sync)	Yes	
Scanning Item	Yes	
Supervisor Features	Yes	Future work planned
Notes + Coin Counting	Yes	
Displaying Totals	Yes	
Get transaction (if voided)	Yes	
Sign Out	NA	

Features Web Application:	Working:	Comment:
Sign In	Yes	
TimeSeries	Yes	Small bug when zooming
Date Picking	Yes	UTC Date Format
Sales Report	Yes	
Stock Report	Yes	
Employee Report	Yes	
File Server	Yes	
Orders	Yes	Future work planned
Till Selection	Yes	
Till Report	Yes	
Sign Out	Yes	

For the web application all of these scenarios were tested across multiple browsers. No issues occurred in any specific browser, besides the minor bug mentioned previous.

Test Plan – EPOS / RetailOnline

The following test plan outlines the guidelines for RELEASE 1, on how both systems should be tested. The system will work under the following given scenarios, anything outside the scope of these scenarios is unlikely to work / break the system.

The point of the following testing is to make sure the system functions as expected. Any bugs found should be recorded after testing. If you feel there is any UX (user experience elements such as textColor, button position etc.) that can be changed please state these as well.

In this given context, you are a qualified retail supervisor / manager, so you have full access to every part of the system for testing purposes.

Scenario 1:

- Customer walks to till with 4 products (on sheet)
- Cashier scans all the products, now the customer has realised they forgot their wallet
- Cashier calls supervisor, supervisor presses the supervisor mode and signs on with their number
- Supervisor presses the “trans void” function in the supervisor menu
- Cashier writes down trans_no in this case (in the real world it would print)
- Till goes back to new sale screen

Scenario 2:

- Customer walks to till with 2 products (any 2 on sheet)
- Cashier scans the products (presses total to get the total amount due)
- Customer hands cashier 10 euro note
- Cashier presses picture of 10 euro note on screen
- System then tells cashier what change to give
- Cashier presses close drawer button (which is usually triggered by the physical cash drawer).
- Transaction is now complete

Before this last scenario, you are going to be shown the exact till total on the online system, this indicates exactly the amount in euro that is in the till in notes and coins. These values will be printed for you then after the transaction is complete, you are going to check these totals to make sure they are correct. (i.e. the right amount of change was deducted)

Scenario 3:

- Customer walks to till with 1 products (any 1 on sheet)
- Cashier scans the products (presses total to get the total amount due)
- Customer hands cashier 5 euro note
- Cashier presses picture of 5 euro note on screen
- System then tells cashier what change to give
- Cashier presses close drawer button (which is usually triggered by the physical cash drawer).
- Transaction is now complete

Scenario 4:

- Customer returns from scenario 1 with their wallet
- Instead of re scanning items the cashier enters the trans_no of that sale
- Cashier presses the total button
- Customer hands cashier 50 euro note
- Cashier presses picture of 50 euro note on screen
- System then tells cashier what change to give
- Cashier presses close drawer button (which is usually triggered by the physical cash drawer).
- Transaction is now complete

User Acceptance Testing (UAT)

The above test plan was executed by ten people in total, some who had never had experience working in retail, others work in retail day in day out and the final group were my work colleagues, these people are very familiar with different types of retail systems.

After the following testing the testers were asked to fill out an anonymous survey, this information was used to see what aspects of the system could be improved for the final release.

Since this isn't a large-scale project, I don't have access to different environments so all test execution is being done locally for the applications and in the cloud for the server.

5.2 Automated Testing

“Automated testing is a method in software testing that makes use of special software tools to control the execution of tests and then compares actual test results with predicted or expected results. All of this is done automatically with little or no intervention from the test engineer. Automation is used to add additional testing that may be too difficult to perform manually.” (28)

Automated testing was only carried out on the website. Throughout this project I used Neotys for testing. It generates every scenario possible, including full load, peak time loads, security testing and also all types of action testing.

After the first release of the website it came back with 376 errors out of 1324 requests. For all the requests the problem was error 404 which is a page not found error. There was a simple solution to this problem, it was a redundant page that I had left in during code migration to the server. So when this page was removed it resolved the error.



Figure 34 Initial Test Results

Additional test results are available in the Appendix section of this report

6. Issues and Risks

As with any big development project, there are always going to be issues and risks to overcome. Typically, they are associated with lack of resources such as time or technologies. The following section outlines the specific issues and risks for this project.

6.1 Issues

Remote Connection

The first issue I encountered, was that when I tried to connect to the remote Microsoft server on google cloud I was getting a connection error when I was connected to eduroam. After almost 2 weeks of triage, I figured out that this was an issue with eduroam, that blocks remote server connections. The solution to this issue was anytime I was working in college I connected to a hotspot on my phone. This allowed me to connect to the server then.

Another issue I faced was on the remote side, using visual studio I couldn't get it to connect to a MySQL database on google cloud. As VS is a Microsoft product by default the database connector built in uses SQL Server which is also a Microsoft product. So, to overcome this issue I will have to install the Oracle MySQL connector for connection. Then in visual studio I will have to install the MySQL tools for Visual Studio, this will enable me to use the "using data.MySQL" directive. At the moment, I haven't resolved the issue but I know how to resolve it moving forward.

Also an issue I faced, is how to store the receipt in the database once the sale is complete. The problem is a receipt id can have multiple products that are the same. I have come up with a solution for this in the database, I will have a receipt lines table, so each line of the receipt will be stored separately in the following format: barcode, PName, Price. So even if the barcodes the same the lines will still be stored. The following ERD explains the solution:

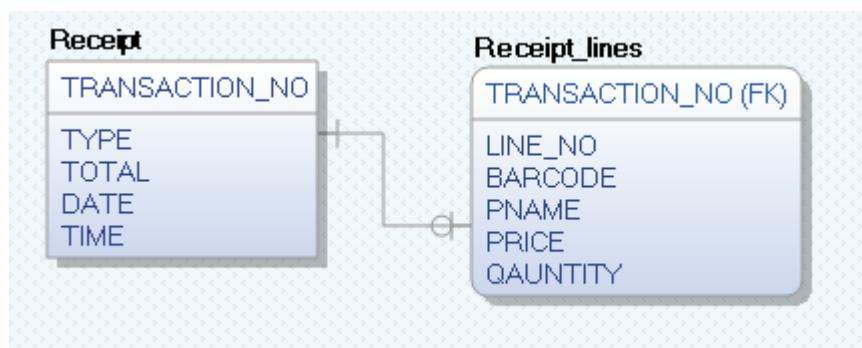


Figure 35 Receipt Lines Table Representation

Parsing

The second issue I encountered was when I was parsing a date_time header file. Formatting differences between Csharp and Java make it very difficult to pass dates between them. As Csharp uses the CultureInfo.InvariantCulture paramater when taking in a date to parse using the DateTime.parseExact() method, the date has to be in this specified format which of course relies on the American localised date_time.

The issue was caused because MySQL only supports dates in the format of “YYYY-MM-DD”. After reading the documentation I found that dates can be manipulated when running a select statement. But when creating/ inserting they have to be in the specified format MySQL has set by default.

“The supported range is '1000-01-01' to '9999-12-31'. MySQL displays DATE values in 'YYYY-MM-DD' format, but permits assignment of values to DATE columns using either strings or numbers.” (29)

To overcome this issue, I had to run a script during runtime which converts from en-US date format to en-GB which supports the date_time format as “DD/MM/YYYY”.

```
ej.addCulture("en-US", { name: "en-US", englishName: "English (United States)" });
ej.addCulture("en-GB", { name: "en-GB", englishName: "English (United Kingdom)",
nativeName: "English (United Kingdom)", numberFormat: { percent: { pattern: ["-n%", "n%"] } },
currency: { pattern: ["-$n", "$n"], symbol: "£" } }, calendars: { standard: { firstDay: 1, patterns:
{ d: "dd/MM/yyyy", D: "dd MMMM yyyy", t: "HH:mm", T: "HH:mm:ss", f: "dd MMMM yyyy HH:mm", F: "dd MMMM yyyy HH:mm:ss", M: "d MMMM" } } } );;
```

Figure 36 Parsing Script

6.2 Risks

An Initial risk with this project was the fact that there are already a number of EPOS systems out there, was this just going to be another generic one? The answer is clearly no, as detailed in the research section of this report this system has so much added functionality that other systems out their lack. To overcome this initial risk during my requirements gathering/ research phase, I looked at other systems out there and explored the pros and cons for each one.

Most systems either run in the cloud or locally, but for a small retailer either of these aren't feasible. If they were to have a local system in their shop this would mean having a physical server onsite. On the other hand, if they were to have a fully cloud based system, what happens if their internet connection goes down. Do they just shut shop? No, my solution is a hybrid of both it allows retailers to take offline sales, once an internet connection is back alive it automatically syncs all their data to the cloud. It's efficient, low cost and a one stop solution.

7. Project Plan

7.1 Future Work

I feel this project has great potential for future work, some day it will be on the market for small retailers. Included in the future work is advanced predictive analysis and fraud detection for notes. These are outlined in detail in the section below.

Reports – Predictive Analysis

Throughout this project I learnt some complex low level parsing of data. Therefore this data is now in the right form to be passed to a machine learning engine such as Microsoft Azure from this models are able to be made.

Below is sample model that I created and can be used in conjunction with the time series in this project to forecast sales for the future.

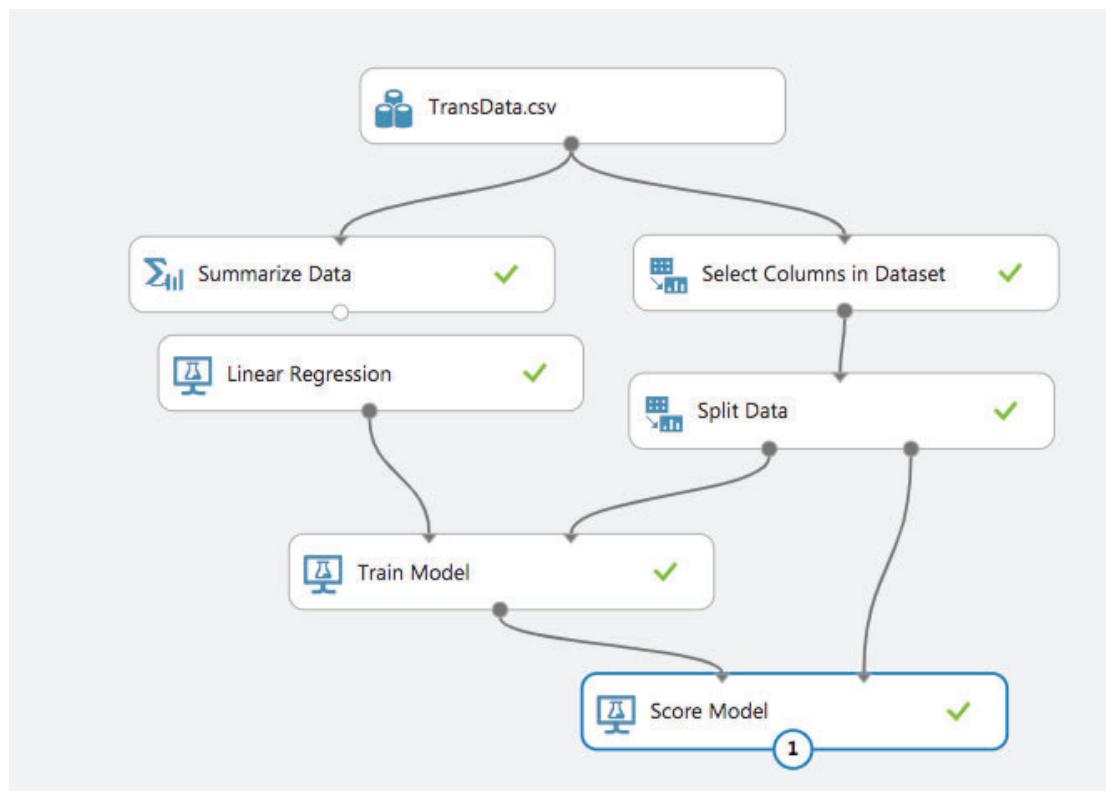


Figure 37 Azure Machine Learning Model

(30)

- First off the data is taken in as a CSV file from the online database in google cloud.
- Then the columns that are going to be used are selected.
- The data is then split 70/30, 70 percent for training the model (in this case using linear regression) and the other 30 percent for scoring the model.
- Using the values from the last year and this year, the model is then able to predict the sales for next year. (roughly 60 percent accuracy).
- This data is then passed back to the google cloud database. The data can then be extracted by the web application for use in the time series.

The model uses a linear regression algorithm, to predict the future data but also model the current data. Regression is the idea of modelling a target value based on independent predictors. It can differ based on the amount of independent variables.

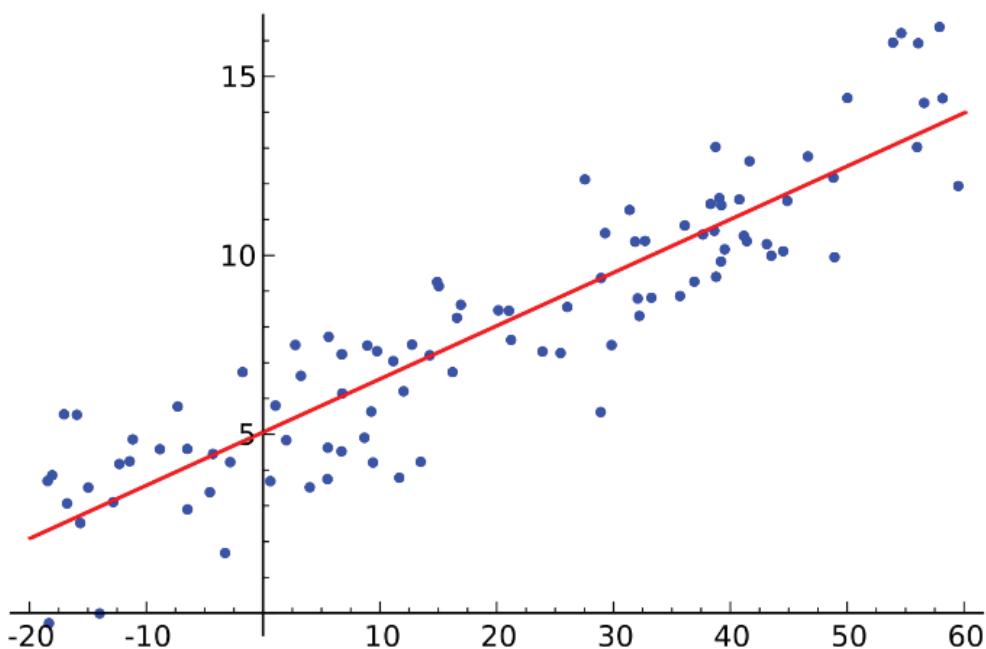


Figure 38 Linear Regression Example

“Linear regression is a type of regression analysis where the number of independent variables is one and there is a linear relationship between the independent(x) and dependent(y) variable.” (31)

$$\text{minimize} \frac{1}{n} \sum_{i=1}^n (pred_i - y_i)^2$$

$$J = \frac{1}{n} \sum_{i=1}^n (pred_i - y_i)^2$$

Figure 39 Cost Function Linear Regression

The cost function helps minimize the error between the predicted value and the actual value. This is to ensure minimal error during the prediction. The easiest way to implement this in Azure is by using R.

As part of this project all the necessary models have been formed in Azure cloud, data has been cleansed and normalised. For this feature the future work is mainly connecting the model to Microsoft web services, that can then connect to the remote web application through visual studio.

7.2 Other Features

Feature:	Local Application / Web Application
Advanced reports including variances	Web Application
Card payments using easypayments api	Local Application
Multi currency – Support for multiple countrys	Both
Machine Learning - Azure	Web Application

The future work for this project is planned to commence in June, when the developer has finished college. The main future work is to include card payments and security handling of these payments under GDPR.

8. Conclusion

Working on this project has given the developer valuable experience, from the initial planning/ research phase of the project to the completed application. Throughout this time the developer has had to react to unforeseen challenges. These challenges have been valuable learning experiences.

The overall aim of this project was to deliver an automated cash counting EPOS system with an online management system. I believe that RetailOnline has delivered the required features in every aspect of the system and has the potential to be brought forward to being very useful software for small and medium retail enterprises.

The developer added unplanned features during the development stage. This was allowed by the methodology employed, which included feature driven development as well as user feedback. These features added greater functionality to the project. These features included:

- File Server
- Timeseries graphing
- Low level parsing – cleansing data

One mistake during development was the developer left all google cloud services running constantly. After the trial credit had been consumed it cost the developer to run these services for the last three months of the project. Next time the developer could run all code locally until the final month and then integrate it onto the server. This highlighted the importance of knowing the limitations of available resources, especially resources not directly under the control of the developer.

If the developer was to undertake a large-scale project like this again, he would spend more time on project planning and design. While it is felt that the methodology selected was largely correct, the developer is now more aware of the importance of time management and planning. This has been a valuable lesson on the role of process in software development.

9. Bibliography

1. Kearney SP, Giordano PA. Method of and system for displaying product related information at POS-based retail checkout systems [Internet]. US8474712B2, 2013 [cited 2018 Dec 4]. Available from: <https://patents.google.com/patent/US8474712B2/en>
2. Chapter 8: Business Processes – Information Systems for Business and Beyond [Internet]. [cited 2019 Mar 30]. Available from: <https://bus206.pressbooks.com/chapter/chapter-9-info-systems-strategic-advantage/>
3. ICRTouch. Retail - ICRTouch EPoS Software [Internet]. Retail - ICRTouch EPoS Software. [cited 2018 Nov 28]. Available from: <https://www.icrtouch.com/sectors/retail>
4. Web_TouchPoint_Brochure.pdf [Internet]. [cited 2018 Nov 28]. Available from: https://www.icrtouch.com/assets/media/docs/Web_TouchPoint_Brochure.pdf
5. 5 reasons 20,000+ retailers love Vend point-of-sale software | Vend [Internet]. [cited 2018 Nov 28]. Available from: <https://www.vendhq.com/why-choose-vend>
6. What is .NET? [Internet]. Microsoft. [cited 2018 Nov 28]. Available from: <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet>
7. MySQL AB. In: Wikipedia [Internet]. 2018 [cited 2018 Nov 28]. Available from: https://en.wikipedia.org/w/index.php?title=MySQL_AB&oldid=867711000
8. History of Java Technology [Internet]. [cited 2018 Nov 28]. Available from: <https://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html>
9. Bootstrap (front-end framework). In: Wikipedia [Internet]. 2018 [cited 2018 Nov 28]. Available from: [https://en.wikipedia.org/w/index.php?title=Bootstrap_\(front-end_framework\)&oldid=870209006](https://en.wikipedia.org/w/index.php?title=Bootstrap_(front-end_framework)&oldid=870209006)
10. Chapter 2 [Internet]. [cited 2018 Nov 28]. Available from: <https://www.w3.org/People/Raggett/book4/ch02.html>
11. How do barcodes and barcode scanners work? [Internet]. Explain that Stuff. [cited 2018 Nov 28]. Available from: <http://www.explainthatstuff.com/barcodescanners.html>
12. Anonymous. EAN/UPC barcodes [Internet]. 2014 [cited 2018 Nov 28]. Available from: <https://www.gs1.org/standards/barcodes/ean-upc>

13. HT 161 HNB Touch Monitor - Hannspree [Internet]. [cited 2019 Mar 27]. Available from: <https://www.hannspree.eu/product/ht-161-hnb/>
14. MySQL :: MySQL 5.6 Reference Manual :: 6.1.2.4 Password Hashing in MySQL [Internet]. [cited 2018 Nov 28]. Available from: <https://dev.mysql.com/doc/refman/5.6/en/password-hashing.html>
15. Home [Internet]. Patents Office. [cited 2019 Mar 31]. Available from: <https://www.patentsoffice.ie/en/>
16. Olli Pitkänen. Legal protection of computer software [Internet]. 05:35:56 UTC [cited 2019 Mar 31]. Available from: <https://www.slideshare.net/ollipitkanen/legal-protection-of-computer-software>
17. Tellermate | Cash Management Solutions [Internet]. Tellermate UK. [cited 2019 Apr 1]. Available from: <https://www.tellermate.com/>
18. Intelligent Cash Drawer | Tellermate LiveDrawer [Internet]. Tellermate UK. [cited 2019 Apr 1]. Available from: <https://www.tellermate.com/products/intelligent-cash-drawer/>
19. Sulato L, Hp W, Yunitasari T. User Requirements Analysis for Restaurant POS and Accounting Application Using Quality Function Deployment. Procedia - Social and Behavioral Sciences. 2015 Jan 20;169:266–80.
20. Feature Driven Development; differences with XP & Scrum [Internet]. Apiumhub. 2017 [cited 2018 Nov 29]. Available from: <https://apiumhub.com/tech-blog-barcelona/feature-driven-development/>
21. tutorialspoint.com. SDLC Waterfall Model [Internet]. www.tutorialspoint.com. [cited 2018 Dec 2]. Available from: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm
22. What is Agile Software Development? [Internet]. Agile Alliance. 2015 [cited 2018 Nov 29]. Available from: <https://www.agilealliance.org/agile101/>
23. User Experience: What is UX? [Internet]. UserTesting Blog. 2015 [cited 2019 Mar 28]. Available from: <https://www.usertesting.com/blog/what-is-user-experience/>
24. What is Functional Testing? - Definition from Techopedia [Internet]. Techopedia.com. [cited 2019 Apr 9]. Available from: <https://www.techopedia.com/definition/19509/functional-testing>
25. What is Regression Testing? | SmartBear Software [Internet]. [cited 2018 Dec 1]. Available from: <https://smartbear.com/learn/automated-testing/what-is-regression-testing/>
26. Interface Testing Tutorial: Types, Strategy & Example [Internet]. [cited 2018 Dec 1]. Available from: <https://www.guru99.com/interface-testing.html>

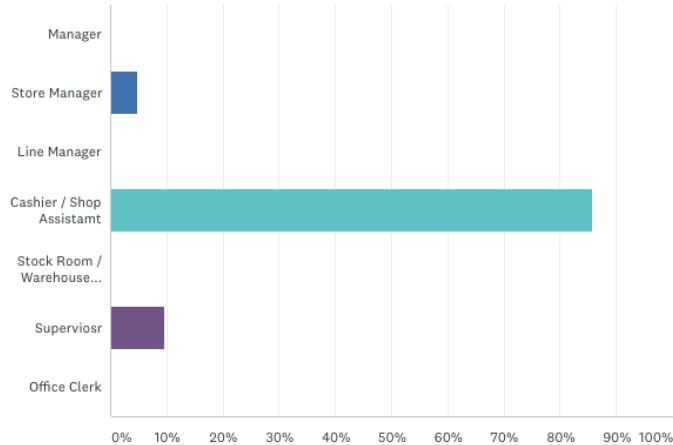
27. tutorialspoint.com. Non-Functional testing [Internet]. www.tutorialspoint.com. [cited 2019 Apr 9]. Available from:
https://www.tutorialspoint.com/software_testing_dictionary/non_functional_testing.htm
28. What is Automated Testing? - Definition from Techopedia [Internet]. Techopedia.com. [cited 2019 Apr 9]. Available from:
<https://www.techopedia.com/definition/17785/automated-testing>
29. MySQL :: MySQL 8.0 Reference Manual :: 11.1.2 Date and Time Type Overview [Internet]. [cited 2019 Apr 9]. Available from:
<https://dev.mysql.com/doc/refman/8.0/en/date-and-time-type-overview.html>
30. Retail | Azure AI Gallery [Internet]. [cited 2019 Apr 5]. Available from:
<https://gallery.azure.ai/industries/retail>
31. Gandhi R. Introduction to Machine Learning Algorithms: Linear Regression [Internet]. Towards Data Science. 2018 [cited 2019 Apr 10]. Available from:
<https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a>

10. Appendix

a. User Requirements Survey

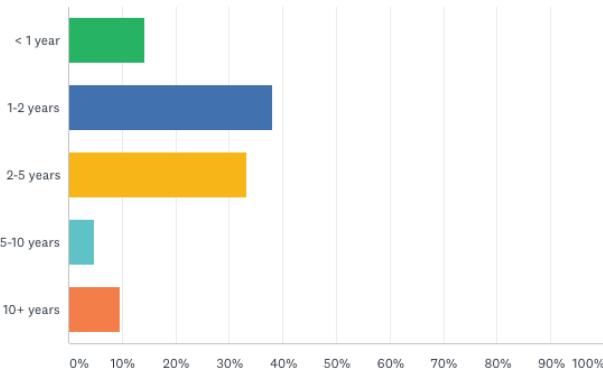
What is your job title/role ?

Answered: 21 Skipped: 0



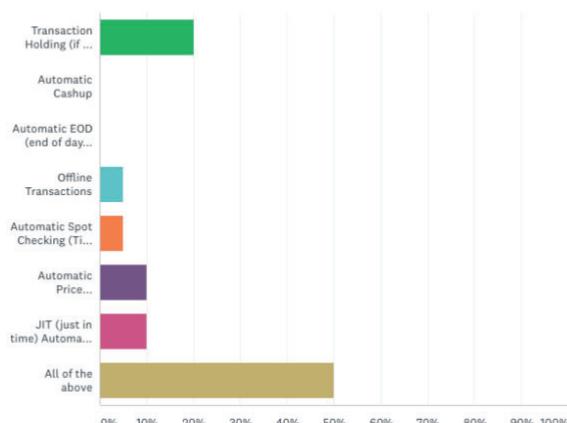
How long have you worked in retail in general ?

Answered: 21 Skipped: 0



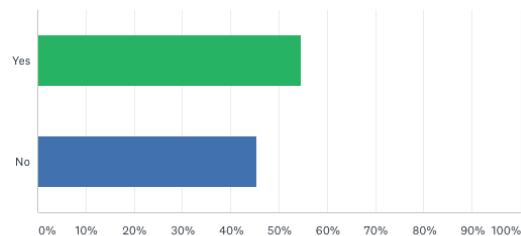
What additional features would you like to see in your systems ?

Answered: 20 Skipped: 1



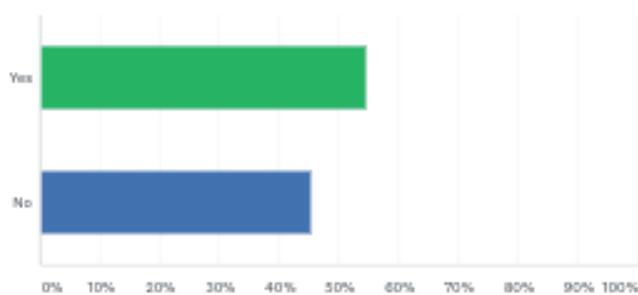
Are you satisfied with your current sales reports ?

Answered: 11 Skipped: 10



Are you satisfied with your current sales reports ?

Answered: 11 Skipped: 10



ANSWER CHOICES	RESPONSES
Yes	54.55%
No	45.45%
TOTAL	11
Comments (5)	

[RESPONSES \(5\)](#) [WORD CLOUD](#) [TAGS \(0\)](#)

Add tags [Filter by tag](#)

Search responses [Q](#) [Reset](#)

Showing 5 responses

[They are unresponsive and not clear](#)

4/9/2019 4:32 PM

[View respondent's answers](#) [Add tags](#)

[We never make budget](#)

2/2/2019 10:17 PM

[View respondent's answers](#) [Add tags](#)

[Don't care](#)

1/2/2019 9:01 PM

[View respondent's answers](#) [Add tags](#)

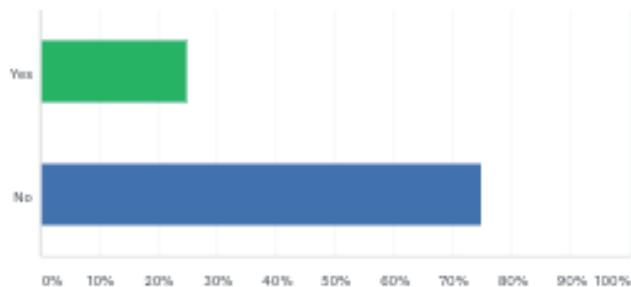
[I don't ever see any](#)

1/2/2019 5:38 PM

[View respondent's answers](#) [Add tags](#)

Does your system give you a good visual representation of all aspects of the business from staff performance to stock monitoring ?

Answered: 12 Skipped: 9



ANSWER CHOICES	RESPONSES
▼ Yes	25.00%
▼ No	75.00%
TOTAL	12

Comments (6)

[RESPONSES \(6\)](#) WORD CLOUD TAGS (0)

Add tags ▾ Filter by tag ▾

Search responses

Showing 6 responses

[Table based reports](#)

4/9/2019 4:32 PM

[View respondent's answers](#) [Add tags ▾](#)

[We can't see current stock levels in other stores easily](#)

2/2/2019 10:17 PM

[View respondent's answers](#) [Add tags ▾](#)

[System is very limited, can't check stock or deliveries](#)

1/3/2019 6:26 PM

[View respondent's answers](#) [Add tags ▾](#)

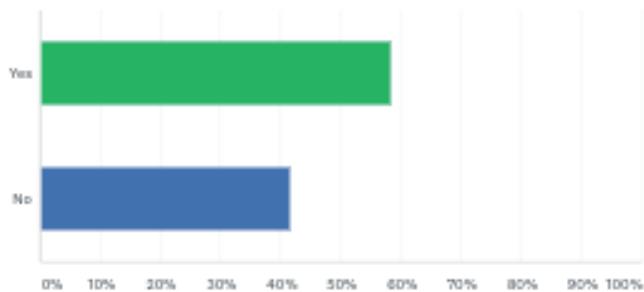
[Nah fam](#)

1/3/2019 9:01 PM

[View respondent's answers](#) [Add tags ▾](#)

Is there a lot of processing time related to inputting new deliveries and updating wrongly priced items ?

Answered: 12 Skipped: 9



ANSWER CHOICES	RESPONSES
▼ Yes	58.33% 7
▼ No	41.67% 5
TOTAL	12

Comments (4)

RESPONSES (4) WORD CLOUD TAGS (0)

Add tags ▾ Filter by tag ▾

Search responses

Showing 4 responses

People actually doing stuff when they say they will.

2/2/2019 10:17 PM

[View respondent's answers](#) [Add tags](#) ▾

Automatic updates Link tills to iPads and computers where stock goes in and out

1/3/2019 6:26 PM

[View respondent's answers](#) [Add tags](#) ▾

Keeping it lit

1/2/2019 9:01 PM

[View respondent's answers](#) [Add tags](#) ▾

Communication through managers

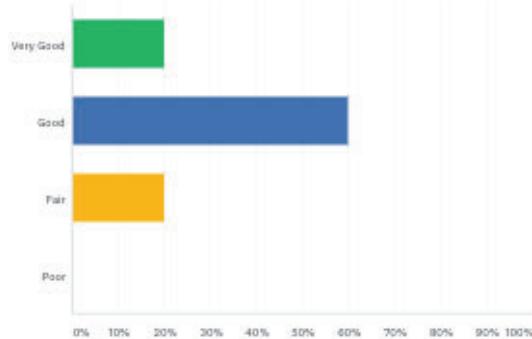
1/2/2019 5:36 PM

[View respondent's answers](#) [Add tags](#) ▾

b. UAT Complete Survey

How useful is this system ?

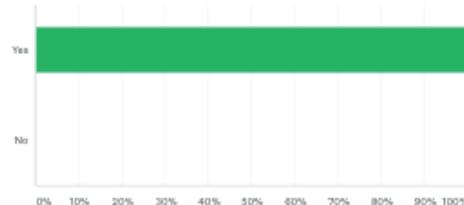
Answered: 5 Skipped: 0



ANSWER CHOICES	RESPONSES
Very Good	20.00%
Good	60.00%
Fair	20.00%
Poor	0.00%
TOTAL	5

Is the totalling, telling the cashier exactly what change to give out useful?

Answered: 5 Skipped: 0



ANSWER CHOICES	RESPONSES
Yes	100.00%
No	0.00%
TOTAL	5

Comments (2)

RESPONSES (2) WORD CLOUD TAGS (0)

PAID FEATURE
Text Analysis lets you search and tag comments and see word clouds of frequent words and phrases. To get this feature, upgrade to a paid plan.

[UPGRADE](#) [Learn more »](#)

Add tags ▾ Filter by tag ▾

Search responses

Showing 2 responses

[It would stop my employees making small errors](#)

4/9/2019 4:27 PM

[View respondent's answers](#) [Add tags](#) ▾

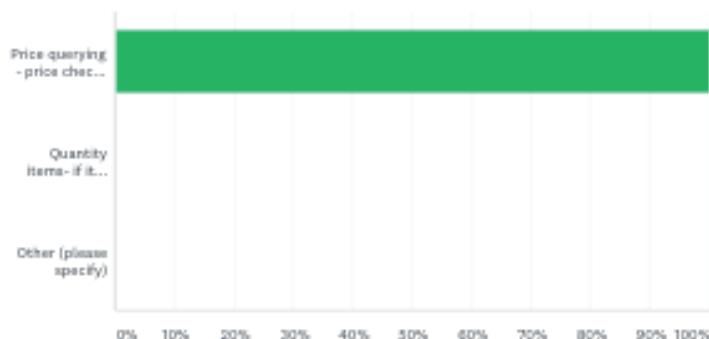
[Eliminates mistakes by younger staff who may not be that familiar with the till](#)

3/28/2019 3:14 PM

[View respondent's answers](#) [Add tags](#) ▾

What new feature would you like to see incorporated into this system ?

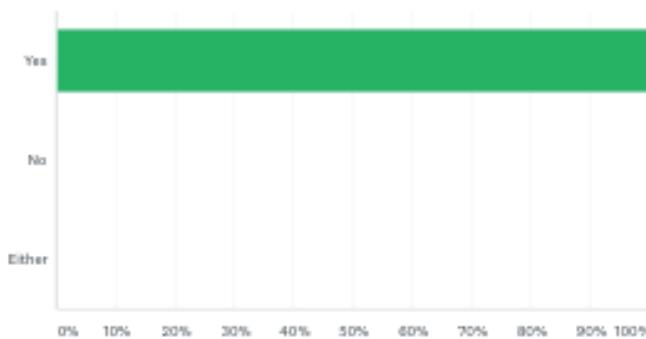
Answered: 5 Skipped: 0



ANSWER CHOICES	RESPONSES
▼ Price querying - price check etc	100.00% 5
▼ Quantity items- if item is the same and customer is buying multiple	0.00% 0
▼ Other (please specify)	Responses 0.00% 0
TOTAL	5

Is data visualization useful in this system?

Answered: 5 Skipped: 0



ANSWER CHOICES	RESPONSES
▼ Yes	100.00% 5
▼ No	0.00% 0
▼ Either	0.00% 0
TOTAL	5

What area do you work/ study?

Answered: 5 Skipped: 0

[RESPONSES \(5\)](#) [WORD CLOUD](#) [TAGS \(0\)](#)

PAID FEATURE

Text Analysis lets you search and tag comments and see word clouds of frequent words and phrases. To get this feature, upgrade to a paid plan.

[UPGRADE](#)

[Learn more >](#)



Apply to selected ▾

Filter by tag ▾

Search responses



Showing 5 responses



Cashier Supermarket

4/9/2019 4:29 PM

[View respondent's answers](#)

Add tags ▾



Retail Store Manager

4/9/2019 4:27 PM

[View respondent's answers](#)

Add tags ▾



Retail Assistant

3/29/2019 1:54 AM

[View respondent's answers](#)

Add tags ▾



retail

3/28/2019 3:14 PM

[View respondent's answers](#)

Add tags ▾

c. Automated Test Results



d. Google Cloud Costs

Mar 1 – 31, 2019		
Documents (2)		
CSV Invoice (1)		
3565269062 (Created: 4/2/19)		
PDF Invoice (1)		
3565269062 (Created: 4/2/19) -		
		Ending balance: €148.73
Date	Description	Amount (EUR)
Mar 1 – 31, 2019	Compute Engine Sustained Usage Discount (Source:Pos System [pos-system-219612])	-€12.02
Mar 1 – 31, 2019	Compute Engine Storage PD Capacity in Frankfurt: 100 Gibibyte-months [Currency conversion: USD to EUR using rate 0.881] (Source:Pos System [pos-system-219612])	€4.23
Mar 1 – 31, 2019	Compute Engine N1 Predefined Instance Ram running in Frankfurt: 2786.249 Gibibyte-hours [Currency conversion: USD to EUR using rate 0.881] (Source:Pos System [pos-system-219612])	€13.40
Mar 1 – 31, 2019	Compute Engine N1 Predefined Instance Core running in Frankfurt: 742.999 Hours [Currency conversion: USD to EUR using rate 0.881] (Source:Pos System [pos-system-219612])	€26.66
Mar 1 – 31, 2019	Compute Engine Licensing Fee for Windows Server 2016 Datacenter Edition (CPU cost): 742.999 Hours [Currency conversion: USD to EUR using rate 0.881] (Source:Pos System [pos-system-219612])	€26.18
Mar 1 – 31, 2019	Cloud SQL D1 usage - hour: 687.083 Hours [Currency conversion: USD to EUR using rate 0.881] (Source:Pos System [pos-system-219612])	€60.53
Mar 1 – 31, 2019	Compute Engine Storage PD Capacity: 49.956 Gibibyte-months [Currency conversion: USD to EUR using rate 0.881] (Source:Pos System [pos-system-219612])	€0.70
Mar 1 – 31, 2019	Compute Engine Storage PD Capacity in London: 9.825 Gibibyte-months [Currency conversion: USD to EUR using rate 0.881] (Source:Pos System [pos-system-219612])	€0.42
Mar 1 – 31, 2019	Cloud SQL Disk usage: 0.972 Gibibyte-months [Currency conversion: USD to EUR using rate 0.881] (Source:Pos System [pos-system-219612])	€0.21
Mar 1 – 28, 2019	Compute Engine Network Internet Egress from Frankfurt to EMEA: 0.521 Gibibytes [Currency conversion: USD to EUR using rate 0.881] (Source:Pos System [pos-system-219612])	€0.06
Mar 1 – 24, 2019	Compute Engine Network Internet Egress from Frankfurt to China: 0.025 Gibibytes [Currency conversion: USD to EUR using rate 0.881] (Source:Pos System [pos-system-219612])	€0.01
Mar 1 – 22, 2019	Compute Engine Network Internet Egress from Frankfurt to Americas: 0.145 Gibibytes [Currency conversion: USD to EUR using rate 0.881] (Source:Pos System [pos-system-219612])	€0.02
Mar 1 – 19, 2019	Cloud SQL External traffic: 0.146 Gibibytes [Currency conversion: USD to EUR using rate 0.881] (Source:Pos System [pos-system-219612])	€0.02
Mar 1 – 10, 2019	Compute Engine Network Internet Egress from Frankfurt to APAC: 0.048 Gibibytes [Currency conversion: USD to EUR using rate 0.881] (Source:Pos System [pos-system-219612])	€0.01
Mar 1 – 4, 2019	Cloud SQL IP address idling - hour: 55.383 Hours [Currency conversion: USD to EUR using rate 0.881] (Source:Pos System [pos-system-219612])	€0.49
Mar 1, 2019	VAT	€31.11
Mar 1, 2019	VAT	-€3.30
Mar 1, 2019	Automatic payment: Visa **** 3131	-€107.16
		Starting balance: €107.16

e. Time Series Javascript

```
90
91
92     <script>
93         $("#legendHeight").val("");
94         $("#legendWidth").val("");
95         var chartObj = $("#Chart1").data("ejChart");
96
97         function alignmentChanged(sender) {
98             $("#Chart1").ejChart("option", { "legend": legend(sender, sender.selectedText) });
99             changeHeightWidth();
100        }
101
102        function positionChanged(sender) {
103            $("#Chart1").ejChart("option", { "legend": legend(sender, sender.selectedText) });
104            changeHeightWidth();
105        }
106        function rowcountChanged(sender) {
107            $("#Chart1").ejChart("option", { "legend": legend(sender, sender.selectedText) });
108            changeHeightWidth();
109        }
110        function columncountChanged(sender) {
111            $("#Chart1").ejChart("option", { "legend": legend(sender, sender.selectedText) });
112            changeHeightWidth();
113        }
114        function legend(sender, data) {
115            switch (sender.model.change) {
116                case "positionChanged":
117                    var legendPosition = data.toLowerCase();
118                    return {
119                        position: legendPosition
120                    }; break;
121                case "alignmentChanged":
122                    var legendAlignment = data.toLowerCase();
123                    return {
124                        alignment: legendAlignment
125                    };
126                    break;
127                case "rowcountChanged":
128                    var count = data;
129                    return { rowCount: count };
130                    break;
131                case "columncountChanged":
132                    var count = data;
133                    return { columnCount: count };
134                    break;
135            }
136        }
137        $('#legendWidth').change(changeHeightWidth);
138        $('#legendHeight').change(changeHeightWidth);
139        function changeHeightWidth() {
140            var width = $("#legendWidth").val();
141            var height = $("#legendHeight").val();
142            var chart = $("#Chart1").ejChart("instance");
143            chart.model.legend.size.width = width;
144            chart.model.legend.size.height = height;
145            chart.redraw();
146        }
147    </script>
```

```
148     <script type="text/javascript">
149     $('#enablehigh').change(function () {
150         var chart = $("#Chart1").ejChart("instance");
151         chart.model.series[0].highlightSettings.enable = $('#enablehigh').is(":checked");
152         chart.redraw();
153     });
154     $('#enablesel').change(function () {
155         var chart = $("#Chart1").ejChart("instance");
156         chart.model.series[0].selectionSettings.enable = $('#enablesel').is(":checked");
157         chart.redraw();
158     });
159     $('#enable3d').change(function () {
160         var chart = $("#Chart1").ejChart("instance");
161         chart.model.enable3D = $('#enable3d').is(":checked");
162         chart.model.enableRotation = true;
163         chart.model.depth = 40;
164         chart.model.wallSize = 2;
165         chart.model.tilt = 0;
166         chart.model.rotation = 34;
167         chart.model.perspectiveAngle = 90;
168         chart.model.sideBySideSeriesPlacement = true;
169         chart.redraw();
170     });
171     $('#enablelege').change(function () {
172         var chart = $("#Chart1").ejChart("instance");
173         chart.model.legend.toggleSeriesVisibility = !$('#enablelege').is(":checked");
174         chart.redraw();
175     });
176     $('#type').change(function () {
177         var chart = $("#Chart1").ejChart("instance");
178         chart.model.series[0].type = $('#type')[0].value;
179         chart.redraw();
180     });
181     $('#highcolor').change(function () {
182         var chart = $("#Chart1").ejChart("instance");
183         chart.model.series[0].highlightSettings.color = $('#highcolor')[0].value;
184         chart.redraw();
185     });
186     $('#highpattern').change(function () {
187         var chart = $("#Chart1").ejChart("instance");
188         chart.model.series[0].highlightSettings.pattern = $('#highpattern')[0].value;
189         chart.redraw();
190     });
191     $('#selcolor').change(function () {
192         var chart = $("#Chart1").ejChart("instance");
193         chart.model.series[0].selectionSettings.color = $('#selcolor')[0].value;
194         chart.redraw();
195     });
196     $('#selpattern').change(function () {
197         var chart = $("#Chart1").ejChart("instance");
198         chart.model.series[0].selectionSettings.pattern = $('#selpattern')[0].value;
199         chart.redraw();
200     });
201     </script>
```