11/3/2017

Finn Turnbull

UNDEE & ANGUS COLLEGE

Inventory Management

Stage 1 – Planning

Contents

[Interpretation of Client Brief 2](#_Toc496868494)

[Aims and Objectives 2](#_Toc496868495)

[Similar Systems 2](#_Toc496868496)

[Project Requirements 3](#_Toc496868497)

[Functional 3](#_Toc496868498)

[Non-Functional 3](#_Toc496868499)

[Resources 4](#_Toc496868500)

[Hardware 4](#_Toc496868501)

[Software 4](#_Toc496868502)

[Diagrams 5](#_Toc496868503)

[Use case 5](#_Toc496868504)

[Class 6](#_Toc496868505)

[Activity 7](#_Toc496868506)

[Data Binding 7](#_Toc496868507)

[User Interface Design 8](#_Toc496868508)

[Wireframes 8](#_Toc496868509)

[ERD 8](#_Toc496868510)

[Normalization 9](#_Toc496868511)

[Sample Table Data 10](#_Toc496868512)

[Project Timeline 10](#_Toc496868513)

[Information Sources and References 10](#_Toc496868514)

# Interpretation of Client Brief

The client, a brewery and distillery who primarily produce gin, whisky and beer wish to upgrade the system which they use to keep track of stock.

From the moment the liquid is produced it must be traced as it is bottled, packaged and readied for sale.  
Currently this is done inefficiently, using multiple systems – some are paper based, others electronic using spreadsheets on Google Drive.

Thus the client wishes to streamline the system to use a single system that staff can update and view easily from anywhere in the workplace.

# Aims and Objectives

The main aim is to securely track gin, whisky and beer from production to sale, using a single system.

This is important for the client, both so that they can keep track of what is in stock, how much there is, and how soon it can be sold, but also to keep track of the duty being paid.  
Proper records must be kept in order to prove to HMRC that tax is being paid on each and every bottle necessary.

It is important that the system can be updated by different departments using laptops located around the facility, so that the data is always kept up to date.

# Similar Systems

There appear to be numerous apps or mobile systems designed to use barcode scanning for inventory control.  
The target market for these kind of programs seem to be small businesses that require a relatively basic system for stock control. (Pontius, 2017)

At the other end of the spectrum there are numerous large scale inventory management applications – a good example of which is Wasp Inventory Control which also relies heavily on scanning barcodes to track stock, eliminating large amounts of manual data entry.  
As well as the software they provide, there are also a number of devices which can be purchased to work with their software, such as a barcode reader, mobile computer and barcode printer.  
This kind of system comes at a cost, but for large businesses this is well worth the outlay.  
 (Inventory Control, n.d.)

# Project Requirements

Functional requirements specify what must be achieved for the system to be considered successfully completed.

Non-functional requirements are optional, and can be added either during the original timeline in spare time, or farther down the road when revisiting development.

## Functional

* Input liquid in volume for whisky, gin and beer,
* For all stock the following must be stored:
  + Name of the gin, whisky or beer type
  + Gyle number
  + The number of bottles
  + Bottle size
  + ABV
  + Location (ie. The bond, warehouse, or cage)
  + Duty stamp range
  + Duty paid or suspended
  + Labelled (Y/N)
* For whisky:
  + Cask type (ie. The wood used)
  + Time spent maturing
* For beer:
  + Storage container used (ie. Pin, firkin, keg, or bottle)
  + Best before date
* Output stock as sales, either as bulk orders or individually

## Non-Functional

* A login system to ensure only employees can access potentially sensitive data,
* Track the supply and use of the vast array of other stock required for production and packaging such as:
  + Casks
  + Bottles
  + Boxes
  + Labels
  + Caps, corks and swing-tops
* Track stock stored in an onsite shop

# Resources

This is the list of hardware and software I will use during development of the system.

To run and use the system a PC or laptop that can run Windows 7 or later will suffice, along with standard peripherals such as a mouse and keyboard to enter data.  
A touch screen would not be optimal, but should still function.

Development of a front end for smartphones and tablets would be useful but shall not be undertaken in this iteration due to time constraints.

## Hardware

A desktop computer provided by D&A College with the following specification:

* Operating system: Windows 7 Professional 64Bit
* Processor: Intel Core i7-4790 @ 3.6GHz

My personal desktop computer with the following specification:

* Operating system: Windows 10 Education 64Bit
* Processor: AMD Ryzen 5 1600 @ 3.2GHz

In both cases a keyboard and mouse are used as input.

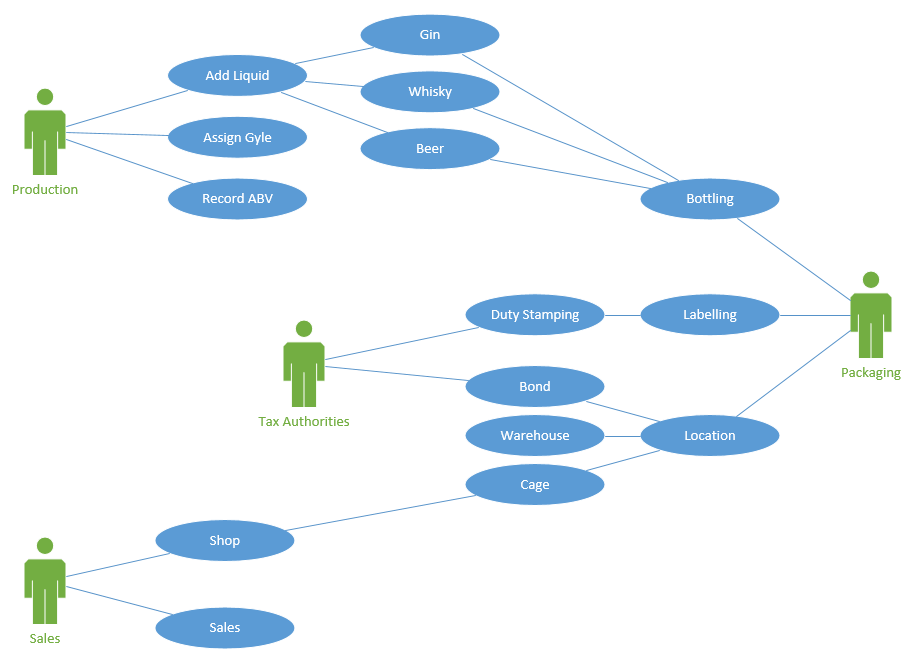
## Software

* Word 2016
* Visio 2016
* Notepad
* Chrome
* Windows Explorer
* Notepad++
* NetBeans
* Visual Studio

# Diagrams

## Use case

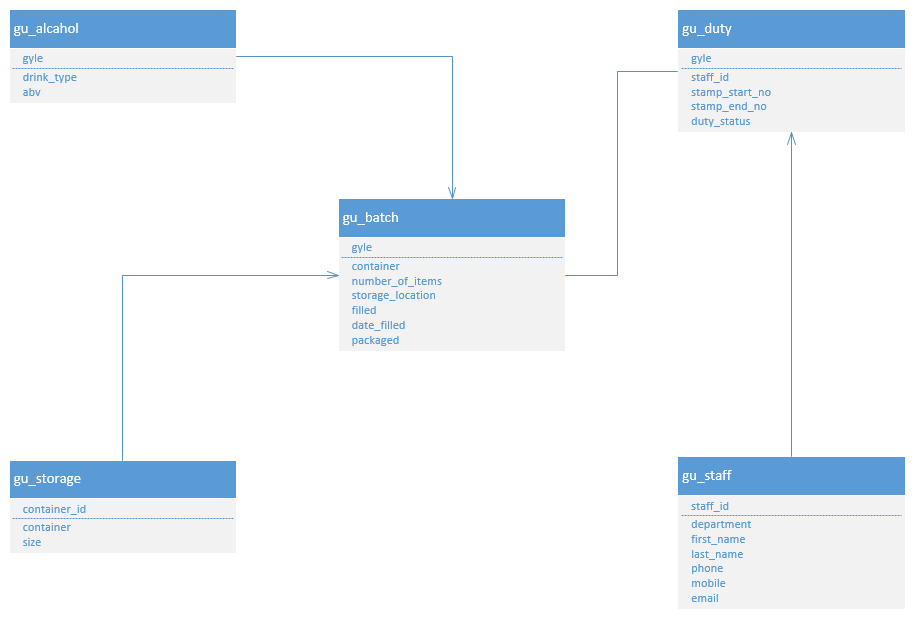
Use case diagrams, or behaviour diagrams demonstrate actions as part of a system(s) and how various actors link to said actions. (UML Use Case Diagrams, n.d.)



## Class

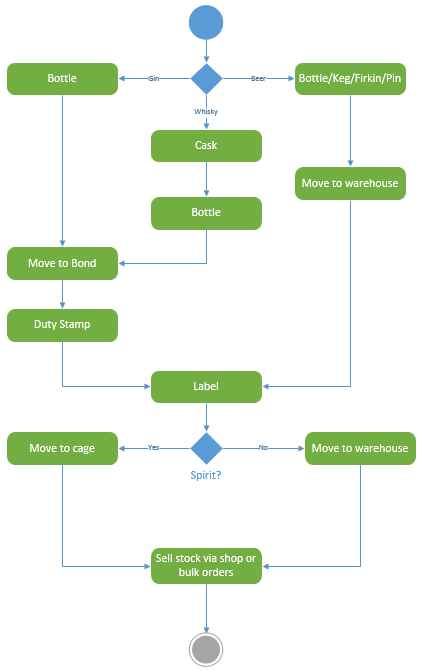
A class diagram depicts a system of classes, their relationships and the attributes of each class.

In this case each class represents a table in a database.



## Activity

An activity diagram depicts the “flow from one operation to another”. (UML - Activity Diagrams, n.d.)



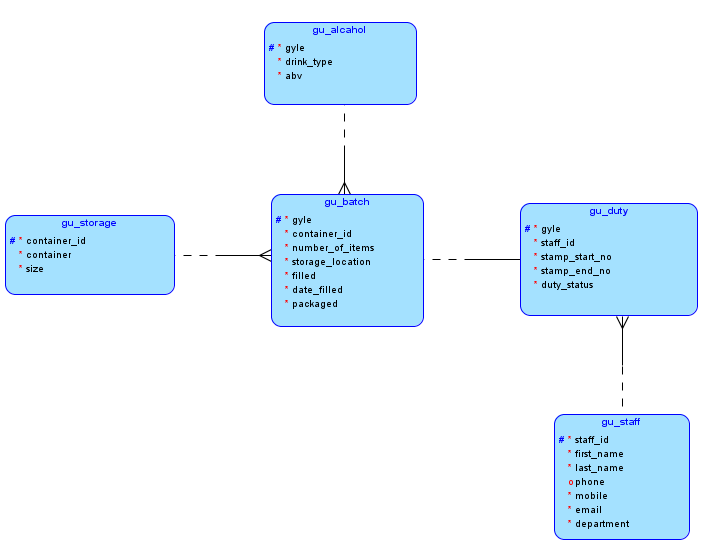
## Data Binding

Research

# User Interface Design

## Wireframes

# ERD



# Normalization

Working from un-normalized form to third normalized form where each table should be simplified to a point where it can be easily implemented.

First normal form requires the fields be grouped together, with a unique identifier called a primary key.  
Fields containing derivable data such as totals must be removed, as they are redundant.

Second normal form only requires changes to be made if the table has a composite key, where the unique key is made up of two non-unique keys.  
If there are composite keys, all non-key fields must rely on all parts of the composite key.

Third normal form eliminates any transitive functional dependency.  
That is to say, there can be no dependency between non-key fields.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UNF | 1NF | 2NF | 3NF | Table Name |
| gyle | #gyle | #gyle | #gyle | gu\_alcohol |
| drink\_type | \*drink\_type | \*drink\_type | \*drink\_type |  |
| abv | \*abv | \*abv | \*abv |  |
| container | \*container | \*container |  |  |
| size | \*size | \*size | #container\_id | gu\_storage |
| number\_of\_items | \*number\_of\_items | \*number\_of\_items | \*container |  |
| filled | \*filled | \*filled | \*size |  |
| date\_filled | \*date\_filled | \*date\_filled |  |  |
| packaged | \*packaged | \*packaged | #gyle | gu\_batch |
| storage\_location | \*storage\_location | \*storage\_location | \*container\_id |  |
| stamp\_start\_no |  |  | \*number\_of\_items |  |
| stamp\_end\_no | #gyle | #gyle | \*storage\_location |  |
| duty\_status | \*staff\_id | \*staff\_id | \*filled |  |
| staff\_id | \*stamp\_start\_no | \*stamp\_start\_no | \*date\_filled |  |
| first\_name | \*stamp\_end\_no | \*stamp\_end\_no | \*packaged |  |
| last\_name | \*duty\_status | \*duty\_status |  |  |
| phone |  |  | #gyle | gu\_duty |
| mobile | #staff\_id | #staff\_id | \*staff\_id |  |
| email | \*first\_name | \*first\_name | \*stamp\_start\_no |  |
| department | \*last\_name | \*last\_name | \*stamp\_end\_no |  |
|  | o phone | o phone | \*duty\_status |  |
|  | \*mobile | \*mobile |  |  |
|  | \*email | \*email | #staff\_id | gu\_staff |
|  | \*department | \*department | \*first\_name |  |
|  |  |  | \*last\_name |  |
|  |  |  | o phone |  |
|  |  |  | \*mobile |  |
|  |  |  | \*email |  |
|  |  |  | \*department |  |
|  |  |  |  |  |

# - Primary key  
\* - Mandatory  
o - Optional

# Sample Table Data

gu\_storage

|  |  |  |
| --- | --- | --- |
| container\_id | container | size (ml) |
| 001 | Ceramic Bottle | 500 |
| 002 | Ceramic Bottle | 700 |

gu\_alcahol

|  |  |  |
| --- | --- | --- |
| gyle | drink\_type | abv (%) |
| L072A | Love Gin | 42.0 |
| E042B | Eden Gin | 42.0 |

gu\_batch

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| gyle | container\_id | number\_of\_items | storage\_location | filled | date\_filled | packaged |
| L072A | 001 | 792 | Warehouse | True | 27/10/17 | True |
| E042B | 002 | 720 | Bond | True | 26/10/17 | False |

gu\_duty

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| gyle | staff\_id | stamp\_start\_no | stamp\_end\_no | duty\_status |
| L072A | 001 | G0002 263 500 | G3002 263 1291 | Paid |

gu\_staff

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| staff\_id | first\_name | last\_name | phone | mobile | email | department |
| 001 | Finn | Turnbull |  | 07020954385 | [slbquista@gmail.com](mailto:slbquista@gmail.com) | Packaging |
| 002 | Joe | Bloggs | 02073779754 | 06420924737 | [example@gmail.com](mailto:example@gmail.com) | Production |

# Project Timeline

# Information Sources and References

*Inventory Control*. (n.d.). Retrieved October 27, 2017, from Wasp Barcode Systems: http://www.waspbarcode.com/inventory-control

Pontius, N. (2017, October 19). *Inventory Management Apps*. Retrieved October 27, 2017, from Camcode: https://www.camcode.com/asset-tags/inventory-management-apps/

*UML - Activity Diagrams*. (n.d.). Retrieved October 27, 2017, from Tutorials Point: https://www.tutorialspoint.com/uml/uml\_activity\_diagram.htm

*UML Use Case Diagrams*. (n.d.). Retrieved October 27, 2017, from UML Diagrams: http://www.uml-diagrams.org/use-case-diagrams.html