Darren Stewart - 538981

FORTH VALLEY COLLEGE |

Graded Unit 2

small business System

SoluTION planning Phase

Contents

[**1 Analysis** 2](#_Toc2540824)

[Business Model Analysis 2](#_Toc2540825)

[1.1 Top level use case diagram 2](#_Toc2540826)

[1.2 Textual Analysis 4](#_Toc2540827)

[1.3 CRC cards 7](#_Toc2540828)

[1.4 Use case descriptions 11](#_Toc2540829)

[1.6 Class diagram and erd 31](#_Toc2540830)

[View Model Analysis 33](#_Toc2540831)

[1.7 Data binding model 33](#_Toc2540832)

[1.8 UI design 33](#_Toc2540833)

[1.9 Client interview 66](#_Toc2540835)

[**2 Design** 68](#_Toc2540836)

[Business model design 68](#_Toc2540837)

[2.1 Dynamic model 68](#_Toc2540838)

[2.1.1 Sequence diagrams 68](#_Toc2540839)

[2.1.2 Activity diagrams 73](#_Toc2540840)

[2.3 Class diagram - finalized 79](#_Toc2540841)

[2.4 ERD - Finalized 80](#_Toc2540842)

[2.4.1 Data table 80](#_Toc2540843)

[View model design 82](#_Toc2540844)

[2.5 Finalized user interface design 82](#_Toc2540845)

[2.6 Data binding model 94](#_Toc2540846)

[Bibliography 104](#_Toc2540847)

# **1 Analysis**

## 

## Business Model Analysis

### 1.1 Top level use case diagram

I have revised the top-level use case diagram that I had made during the inception phase to include more detail It now shows the each of the uses and other use case that branches from the main node as well as showing this top level use case I will be also including sub diagrams that focus on a specific use case going into more detail showing what information each use case will need to take in the reason that I am doing this is to keep the diagram for getting too complicated to read.

I have decided to move some of the use cases into different sections of the application as I feel that the use cases are clearer with the new sections that they were added to.

The purpose of using this diagram is to help me identity what functions the system is required to carry out during the inception phase of the project I created a list of functional and non-functional requirements I used those functional requirements to help create the diagram as each functional requirement represents one of the use cases in the diagram.



### 1.2 Textual Analysis

I have decided to perform a textual analysis of the brief that was given to me by the client the reason that I performed a textual analysis of the brief is to identify any nouns and verbs in the brief. The way that I have done the analysis is to highlight the nouns and verbs in different colours so that I am able to clearly identify them

The reason that carrying out a textual analysis is because the nouns are potential the potential entities that will be in the system, I have highlighted the nouns green While the verbs are actions that the system are required to carry out.

Textual analysis is a useful analysis technique to help identify the actions that the system is required to carry out and what entities will make up the system.

After going through the brief a few times picking out the nouns and verbs I have them separated them into separate columns to summaries what I had found during the analysis.

**Brief**

I have been approached by a small game retailer that is based in the uk who are looking for a software developer to build them an application to help them expand their business.

The application that the company wants build is a small business application with a wide range of features that help them track important details about how their business is performing. The features that the company want to be included in the application is that is for the application to be able to cover sales invoicing and for them to be able to control their stock in real time. The company wants the application to produce a sales invoice for each transaction and to update their stock in real-time.

The application should display information on screen and to also be able to produce paper reports that detail their daily, weekly, monthly, quarterly and annual sales analysis, VAT analysis, stock turnover and stock outages. The company wants to also be able to display customer, product, supplier and invoice information they want this to include transaction logs to have an option to print any of the transaction logs.

They want the application to be able to work on computers with low specifications running windows xp, Linux or Mac OS X or newer the application should either be a console-based application or have a graphical user interface.

**Entities**

|  |  |
| --- | --- |
| **Entity name** | **Description** |
| Small game retailer | The people who I am making the application for and the people who will use the application |
| Graphical use interface | The interface that users will use to interact with the application |
| Sales reports | The different types of reports that the application will need to produce |
| Transaction | Employees will be able to produce transactions in the system |
| Transaction logs | Each of the transactions made will produce a transaction log which will hold the details about the transaction made |
| Small business application | The thing that I have been asked to build for the company |
| Sales invoice | The rescript made after a transaction has been made with the company it will hold all the sales information made |
| Customer Information | Information that is stored about the customers the required information to be stored will include their first name, last name, address, email address and give the user the option to give their phone number more information to be stored may be identified at a later stage |
| Product information | Information that is stored about the products the required information to be stored will include the product name, product cost, quantity in stock more information may be identified in a later stage |
| Supplier information | Information that is stored about the company’s suppliers the information that will be stored is the supplier name and address more information may be identified at a later stage. |
| Invoice Information | Information that is stored about the invoices that the company produces the information that will be stored is the name of the person who made the purchase and the name of the product they purchased more information may be identified at a later stage |

**Actions**

|  |  |
| --- | --- |
| **Action Name** | **Description** |
| Track important details | The application will be required to store important information about their customer, product, suppliers and invoices |
| Produce sales invoice | After each sale the application will produce an invoice that will contain information about the purchase made |
| Produce reports | The application is required to produce a range of different sales reports and reports about their stock levels |
| Display information | The application will be able to display information about their customers, products, suppliers and invoices. |
| Update stock in real-time | The application will have a function that lets an employee update the stock level of an item that they have stored in the system |
| Print reports | Any of the reports that have been generated by the application will have the option to be printed. |

**Actors**

|  |  |
| --- | --- |
| **Actor Name** | **Description** |
| Database | The main storage structure that is being used in the application any information that is being saved in the application will be stored in the data base and any information being accessed will likely be retrieved from a table in the data base |
| Employees | The people who will be the main users of the application. Each employee will have unique log in details to sign into the system. |

### 1.3 CRC cards

#### 1.3.1 CRC cards - ERD

I am using crc cards to help me identify the entities that will make up the erd for the data base I am also using them to help identify any of the attributes as well as a brief description about what the attribute will store.

|  |  |
| --- | --- |
| **Entity**: employees | |
| **Attributes**   * First Name * Last Name * Password * Employee id | **Description**   * the employees first name * The employees last name * Stores the password that is assigned to an employee * Unique Id that can be used to identify an employee |
|  |  |

|  |  |
| --- | --- |
| **Entity**: Product | |
| **Attributes**   * Product Name * Quantity in stock * Cost of unit * Release date * Image of product * The platform * Description * Product ID | **Description**   * The name of the product * How much of a product is available? * The date that the product was released * An image of the product * The platform the product was released on * A brief description of the product * Unique Id that can be used to identify a product |
|  |  |

|  |  |
| --- | --- |
| **Entity**: Customer | |
| **Attributes**   * First Name * Last Name * Address * Email Address * Phone Number * Customer id | **Description**   * The first name of the customer * The last name of the customer * The customers address * The customers email address * Optionally allow the customer to store their phone number * Unique Id that can be used to identify a Customer |
|  |  |

|  |  |
| --- | --- |
| **Entity**: Supplier | |
| **Attributes**   * Supplier Name * Supplier address * Supplier id | **Description**   * The name of the supplier * The address of the supplier * Unique Id that can be used to identify a supplier |
|  |  |

|  |  |
| --- | --- |
| **Entity**: Invoice | |
| **Attributes**   * Customer first name * Customer last name * Customer address * Customer Email address * Name of Product bought * Quantity sold * Cost of product * Invoice ID | **Description**   * The first name of the customer * The last name of the customer * The address of the customer * The email address of the customer * The name of the product that they bought * The cost of the product they bought * Unique Id that can be used to identify a supplier |
|  |  |

#### 1.3.2 CRC cards - Classes

I have created a list of crc cards to help me identify the classes that will be used in the program and what their purpose is within the program and what other classes they will interact with.

|  |  |
| --- | --- |
| **Class:** User Interface | |
| **Responsibility**   * Allow user to interact with the application * Display information to screen | **Collaborators**   * Data base * User interface * Printer |
|  |  |

|  |  |
| --- | --- |
| **Class:** Database connector | |
| **Responsibility**   * Allows the application to interact with the data base * Information in the data base can be saved | **Collaborators**   * User Interface * Main |
|  |  |

|  |  |
| --- | --- |
| **Class:** Printer | |
| **Responsibility**   * Allows the user to print reports generated by the program | **Collaborators**   * User Interface * main |
|  |  |

|  |  |
| --- | --- |
| **Class:** Employee | |
| **Responsibility**   * Holds information about the employee from the data base * Interacts with the data base to retrieve information from the data base and stores it in the program so the information can be validated | **Collaborators**   * User Interface * Main * Data base |
|  |  |

|  |  |
| --- | --- |
| **Class:** Customer | |
| **Responsibility**   * Holds information about the customer from the data base * Interacts with the data base to retrieve information from the data base about the customer so the information can be validated | **Collaborators**   * User Interface * Main * Data base |
|  |  |

|  |  |
| --- | --- |
| **Class:** Invoice | |
| **Responsibility**   * Holds information about the invoice from the data base * Interacts with the data base to retrieve information from the data base about the invoice so the information can be validated | **Collaborators**   * User Interface * Main * Data base |
|  |  |

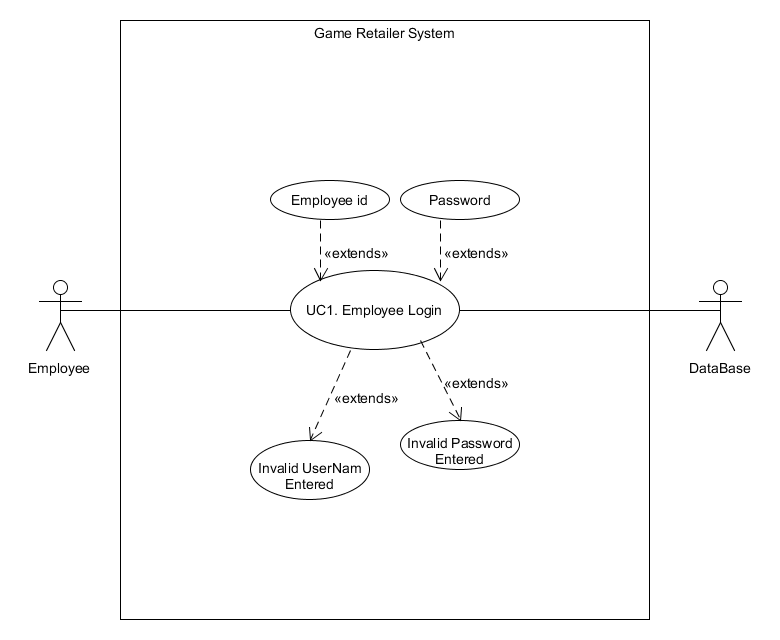
|  |  |
| --- | --- |
| **Class:** Product | |
| **Responsibility**   * Holds information about the product from the data base * Interacts with the data base to retrieve information from the data base about the products so the information can be validated | **Collaborators**   * User Interface * Main * Data base |
|  |  |

### 1.4 Use case descriptions

After creating the top-level use case diagram (section 1.1), I then went on to create a more detailed list of use case discerptions that go into a more detailed on how each of the use cases could potentially function in the application. Doing these use cases gives me a good idea of what users and the system will need to do to carry out sections of the application the use cases will also show what will happen if users give invalid information and how that will be dealt with.

As well as giving an idea of how the process will function it will also give me an idea of what condition needs to be met for the use case to be carried out and what will happen when the use case is completed and how the user will be able to start the use case.

#### 1.4.1 employee log in



**Name**:  employee Login

**Primary Actor(s)**:  employee

**Description**:

Whenever the user loads up the application, they will be required to log in they must have valid log in. Details that are stored in the system. The log in will consist of their employee number and a password.

**Post-conditions**:  employee is logged into system and the main menu of the application is displayed.

**Trigger event:** Automatically loads to this page when the application is started

**Normal flow**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| NF-1 | the system prompts the user to log in to the system |  |
| NF-2 | Employee enters their details |  |
| NF-3 | system verifies that the user name is correct | refer AF-A |
| NF-4 | system verifies that the password is correct | refer AF-B |
| NF-5 | user is logged into system |  |
| NF-6 | Welcome screen is displayed |  |

**Alternative flow**

**AF-A**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF A-1 | Incorrect user name is entered |  |
| AF A-2 | Error message is displayed | Refer EM-1 |
| AF A-3 | System returns to NF-1 |  |

**AF-B**

|  |  |  |
| --- | --- | --- |
| **Step Number** | Action | Reference |
| AF B-1 | Incorrect password is entered |  |
| AF B-2 | Error Message is displayed | Refer EM-2 |
| AF B-3 | System returns to NF-1 |  |

**Error messages**

|  |  |
| --- | --- |
| Error message number | Action |
| EM -1 | Invalid user name entered please enter a valid one |
| EM – 2 | Invalid password entered please enter a valid one |

#### 1.4.2 Use Case 2 - Produce sales invoice



**Name**:  Produce sales invoice

**Primary Actor(s)**:  employee

**Description**:

After each sale the employee will be able to generate a sales invoice the invoice will show the name of the product that was purchased the first and last name of the person who made the purchase their address, email address the cost of the product they bought, and the application will assign an invoice id to the invoice.

**Pre-conditions**:  employee is logged into the system

**Post-conditions**:  Sales invoice is produced

**Trigger event:** Clicking on the create sales invoice button

***Scenario***

***Normal Flow***

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| NF-1 | System prompts employee to enter information about the sale   * Customer id * Product id * Quantity sold | Refer AF-A |
| NF-2 | System checks stock level of the product | Refer AF-B |
| NF-3 | System confirms the information is valid |  |
| NF-4 | System assigns an invoice id to the invoice |  |
| NF-5 | Invoice is saved to the data base |  |

**Alternative Flows**

**AF-A**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-A-1 | invalid data was entered |  |
| AF-A-2 | error message is displayed | EM-1 to EM-3 |
| AF-A-3 | Return to NF-1 |  |

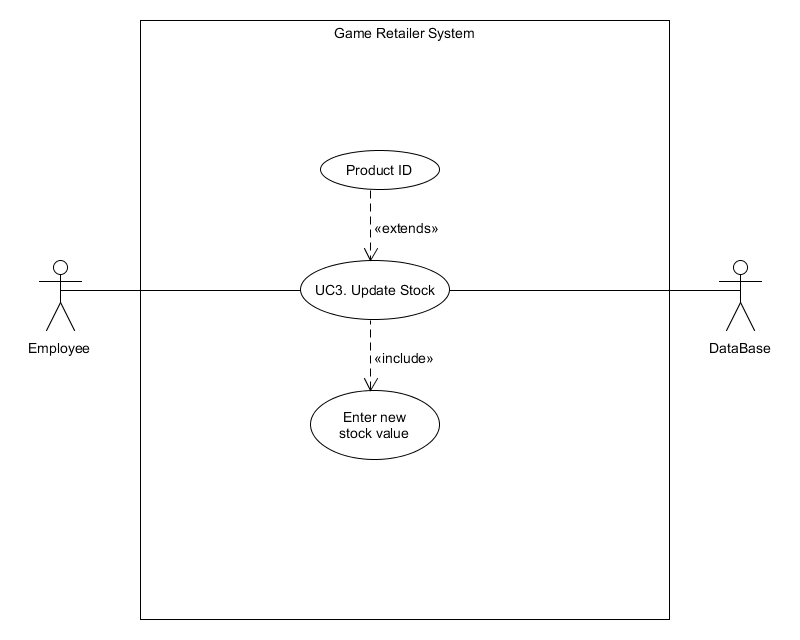
**AF-B**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-B-1 | Stock level is too low for the purchase |  |
| AF-B-2 | Error Message is displayed | EM-4 |
| AF-B-3 | Return to NF-1 |  |

**Error Messages**

|  |  |
| --- | --- |
| **Error message number** | **Action** |
| EM-1 | Customer id not found please enter a valid Customer id |
| EM-2 | Product id not found please enter a valid product id |
| EM-3 | Invalid Quantity entered please enter a valid one |
| EM-4 | Not enough of the product in stock |

#### 1.4.3 Use Case 3 – update stock



**Name**:  update stock

**Primary Actor(s)**:  employee

**Description**:  The user will be able to update the stock that they sell they will be able to do this by updating the information in the database using the program.

**Pre-conditions**:  employee is logged into the system

**Post-conditions**:  the stock will be updated

**Trigger event:** employee presses the update stock button

**Normal flow**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| NF-1 | System prompts the user to enter the product id | Refer AF-A |
| NF-2 | User enters product id |  |
| NF-3 | System displays the current stock level of the product |  |
| NF-4 | System prompts user to enter the new stock level |  |
| NF-5 | New stock level is entered | AF-B-2 |
| NF-6 | New stock level is updated in the data base |  |
| NF-7 | Change is saved to data base |  |

**Alternative flow**

AF-A

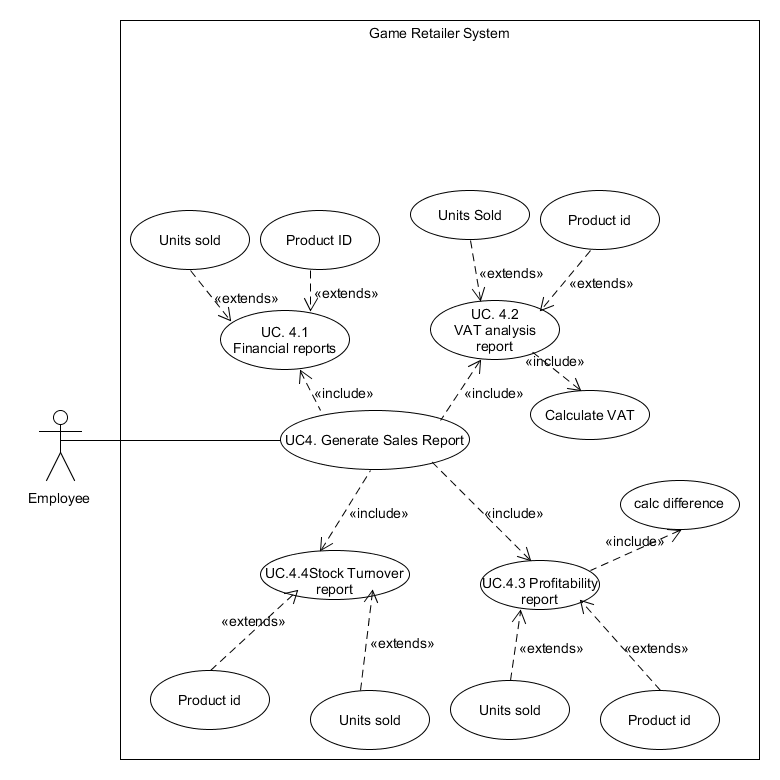
|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-A-1 | Invalid product id was entered |  |
| AF-A-2 | Display Error Message | EM-1 |
| AF-A-3 | Return to NF-1 |  |

AF-B

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-B-1 | Invalid stock level was entered |  |
| AF-B-2 | Display error message | EM-2 |
| AF-B-3 | Return to NF-4 |  |

|  |  |
| --- | --- |
| **Error message number** | **Action** |
| EM-1 | Product id not found please enter a valid product id |
| EM-2 | Stock level invalid please enter a valid stock level |

#### 1.4.4 Use Case 4 – Generate Sales Report



**Name**: Generate Sales report

**Primary Actor(s)**:

**Description**:  The user will be able to produce a range of reports detailing their daily, weekly, monthly, quarterly, annual sales, profitability and vat analysis

**Pre-conditions**:  employee is logged into the system

**Post-conditions**:  Sales report is generated and displayed to screen

**Trigger event:** employee presses the generate sales report button

***Scenario***

**Normal Flow**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| NF-1 | System prompts the user to select the report type they want to generate |  |
| NF-2 | User selects the report that they want to generate | Refer AF-A financial report  Refer AF-B VAT total  Refer AF-C profitability report |
| NF-3 | information required for the report is generated | Refer report display information |
| NF-4 | Report is displayed to screen |  |

**Financial report**

**AF-A**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-A-1 | User selects to generate a financial report |  |
| AF-A-2 | System prompts user to enter the product id of the product they want a report of | Refer AF-AE |
|  | System prompts the user to the price the unit was purchased for | Refer AF-BE |
| AF-A-3 | System calculates the total sales of the product |  |

**AF-AE**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-AE-1 | User enters an invalid product id |  |
| AF-AE-2 | Display Error Message | EM-1 |
| AF-AE-3 | Return to NF-1 |  |

**AF-BE**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-BE-1 | User enters an invalid price |  |
| AF-BE-2 | Display Error Message | EM-3 |
| AF-BE-3 | Return to NF-1 |  |

**VAT analysis**

**AF-B**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-B-1 | User selects to generate a VAT analysis |  |
| AF-B-2 | System prompts user to enter the product id | Refer to AF-AE |
| AF-B-3 | System prompts user to enter the number of units sold | Refer to AF-BE |
| AF-B-4 | System Calculates VAT of the products |  |

**AF-AE**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-AE-1 | User enters an invalid product id |  |
| AF-AE-2 | Display Error Message | EM-1 |
| AF-AE-3 | Return to NF-1 |  |

**AF-BE**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-BE-1 | User enters an invalid product id |  |
| AF-BE-2 | Display Error Message | EM-2 |
| AF-BE-3 | Return to NF-1 |  |

**Profitability**

**AF-C**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-C-1 | User selects to generate a stock turnover report |  |
| AF-C-2 | System prompts user to enter the product id | Refer to AF-AE |
| AF-C-3 | System prompts user to enter the number of units sold | Refer to AF-BE |
| AF-C-4 | System calculates the price difference |  |

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-AE-1 | User enters an invalid product id |  |
| AF-AE-2 | Display Error Message | EM-1 |
| AF-AE-3 | Return to NF-1 |  |

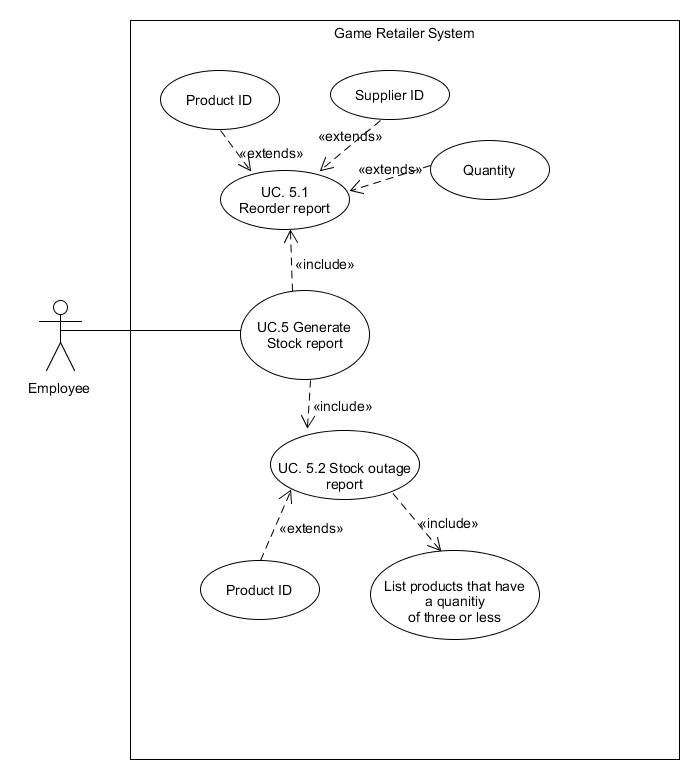
|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-BE-1 | User enters an invalid product id |  |
| AF-BE-2 | Display Error Message | EM-2 |
| AF-BE-3 | Return to NF-1 |  |

|  |  |
| --- | --- |
| **Error Message Number** | **Error message** |
| EM-1 | Product id not found please enter a valid product id |
| EM-2 | Invalid number of units sold entered please enter a valid number |
| EM-3 | Invalid price entered please enter a valid one |

**Report information**

|  |  |
| --- | --- |
| **Report type** | **Displayed Information** |
| Daily sales | product Id, units sold, cost of unit, Vat, total |
| Weekly sales | product Id, units sold, cost of unit, Vat, total |
| Monthly sales | product Id, units sold, cost of unit, Vat, total |
| Quarterly Sale | product Id, units sold, cost of unit, Vat, total |
| Annual Sales | product Id, units sold, cost of unit, Vat, total |
| Vat analysis | product Id, cost of unit, cost of unit with vat, vat total |
| Profitability | purchase price, sale price, difference |

#### 1.4.5 Use Case 5 – Generate Stock Report



**Name**:  Generate Stock report

**Primary Actor(s)**:  employee

**Description**: The user will be able to generate reports about their stock turnover and stock outages.

**Pre-conditions**:  employee is logged into the system

**Post-conditions**:  stock report is generated

**Trigger event:** employee presses the generate stock report button

***Scenario***

***Normal Flow***

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| NF-1 | System prompts the user to select the report type they want to generate |  |
| NF-2 | Report type to be generated is selected |  |
| NF-3 | system prompts employee to enter information Based on the report that was selected |  |
| NF-4 | information required for the report is entered | refer to required information and (AF A-1) |
| NF-5 | information required in the display of the report is generated |  |
| NF-6 | Report is displayed to screen | refer to report information displayed |

**Alternative flow**

**AF-A**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-A-1 | Invalid product id entered |  |
| AF-A-2 | Display error message | EM-1 |
| AF-A-3 | Return to NF-4 |  |

**AF-B**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-B-1 | Invalid Total units sold entered |  |
| AF-B-2 | Display error message | EM-2 |
| AF-B-3 | Return to NF-4 |  |

**AF-C**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-C-1 | Invalid supplier id entered |  |
| AF-C-2 | Display error message | EM-3 |
| AF-C-3 | Return to NF-4 |  |

**AF-D**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-D-1 | Invalid order quantity |  |
| AF-D-2 | Display error message | EM-4 |
| AF-D-3 | Return to NF-4 |  |

**Error messages**

|  |  |
| --- | --- |
| **Error message number** | **Error message** |
| EM-1 | Invalid product id entered please enter a valid value |
| EM-2 | Invalid total units sold entered please enter a valid value |
| EM-3 | Invalid supplier id please enter a valid value |
| EM-4 | Invalid order quantity entered please enter a valid value |

***Extension Flows***

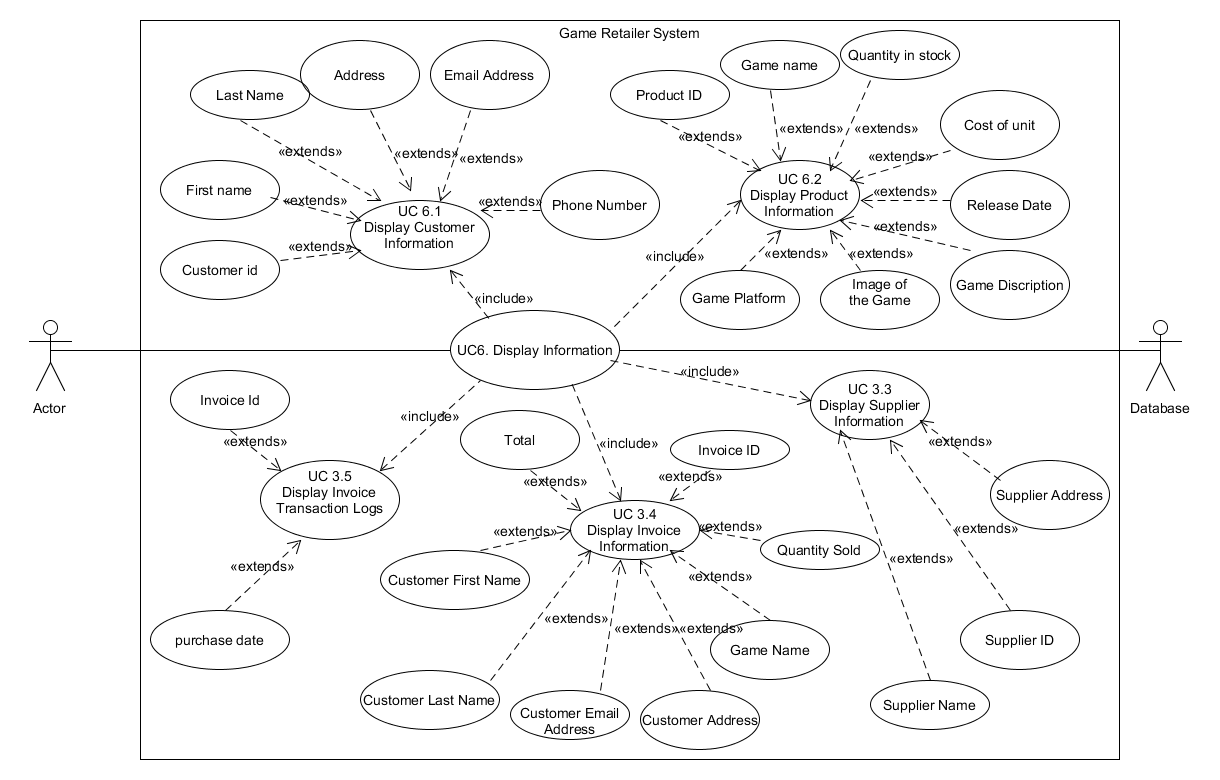
**Report required information**

|  |  |
| --- | --- |
| **Report type** | **Required information** |
| Stock outage report | Product Id |
| Stock turnover | Product id and total unit sold |
| Reorder report | supplier id product id, quantity to order |

**Report information displayed**

|  |  |
| --- | --- |
| **Report type** | **Displayed Information** |
| Stock outage report | product id, product name of products that have a quantity of zero |
| Stock turnover report | Product id, product name, total units sold |
| Reorder report | supplier id product id, quantity to order |
|  |  |

#### Use Case 6 – display information



**Name**:  display information

**Primary Actor(s)**:  employee

**Description**:  The user will be able to display a list of information that is stored in the data base about their customers, products, suppliers and invoices.

**Pre-conditions**:  employee is logged into the system

**Post-conditions**:  details that the employee wants will be displayed to the screen

**Trigger event:** user presses the display information button

***Scenario***

***Normal Flow***

|  |  |  |
| --- | --- | --- |
| **Step Number** | Action | Reference |
| NF-1 | system prompts the user to select what information they want displayed   * Customer * Product * Supplier * Invoices |  |
| NF-2 | employee selects what they want to view | (Refer to AF-A for customer, AF-B for product, AF-C for suppliers and AF-D for invoices) |

**Alternative flow**

**AF-A**

|  |  |  |
| --- | --- | --- |
| **Step Number** | Action | Reference |
| AF-A-1 | user selected display customer information |  |
| AF-A-2 | Retrieve customer information from data base |  |
| AF-A-3 | system displays list of customers details   * Customer Id * First name * Last name * Address * Email address * Phone number (optional) |  |

**AF-B**

|  |  |  |
| --- | --- | --- |
| **Step Number** | Action | Reference |
| AF-B-1 | user selected to display product details |  |
| AF-B-2 | Retrieve product information from data base |  |
| AF-B-3 | system displays list of product details   * Product name * Quantity in stock * Cost of unit * Release date * An Image of the product * The platform * Description |  |

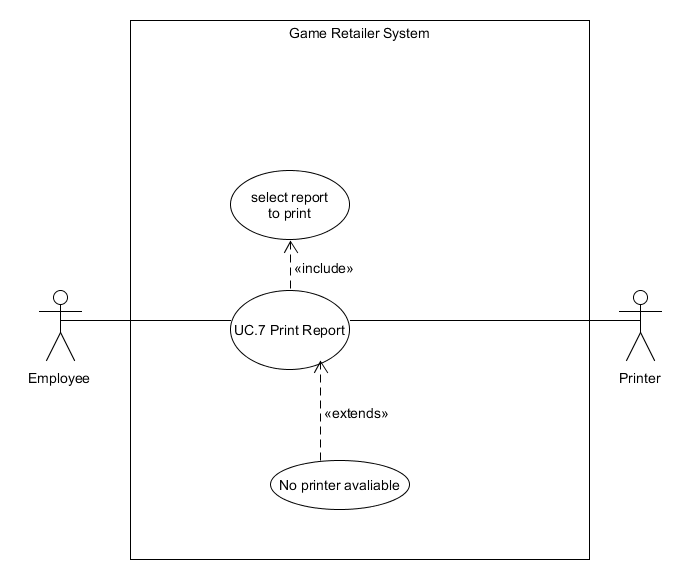
**AF-C**

|  |  |  |
| --- | --- | --- |
| **Step Number** | Action | Reference |
| AF-C-1 | user selected to display supplier details |  |
| AF-C-2 | System retrieves supplier information from data base |  |
| AF-C-3 | system displays list of supplier details   * Supplier name * Supplier Address * Supplier id |  |

**AF-D**

|  |  |  |
| --- | --- | --- |
| **Step Number** | Action | Reference |
| AF-D-1 | user selected to display list of invoices |  |
| AF-D-2 | System retrieves invoice information from data base |  |
| AF-D-3 | system displays list of invoice details   * Customer first name * Customer last name * Customer address * Customer email address * Name of the product they bought * Quantity sold * Cost of the product * Invoice id |  |

#### 1.4.7 Use Case 7 – print report



**Name**:  Print report

**Primary Actor(s)**:  employee

**Description**:  The user will have the option for any of the reports that have been generated by the application to be printed out.

**Pre-conditions**:  employee is logged into the system

**Post-conditions**:  report that the user wants to print is printed

**Trigger event:** employees presses the print report button

***Scenario***

***Normal Flow***

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| NF-1 | system prompts the user to select what report they want to print |  |
| NF-2 | System checks if there is an available printer | Refer AF-A |
| NF-3 | system finds printer that is available |  |
| NF-4 | report is printed |  |

**AF-A**

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Action** | **Reference** |
| AF-A-1 | system couldn’t find a printer |  |
| AF-A-2 | display error message | Refer EM-1 |
| AF-A-3 | return to NF-1 |  |

**Error message**

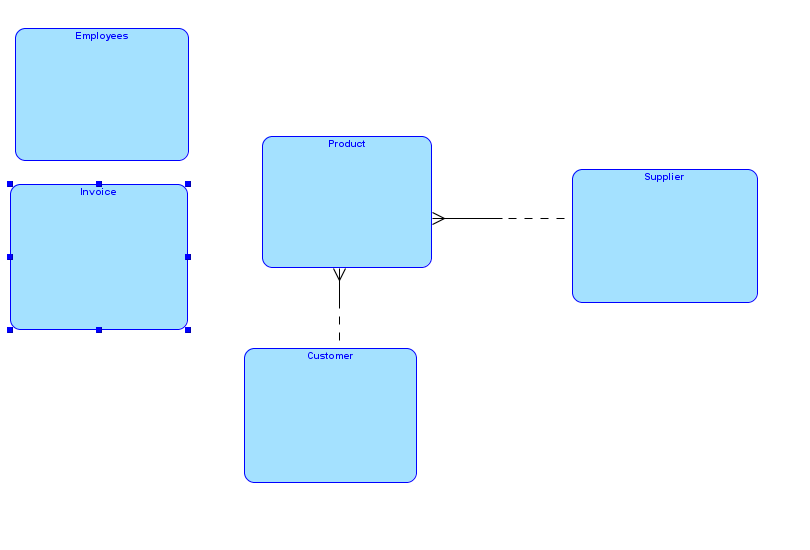
|  |  |
| --- | --- |
| **Error Message Number** | **Error Message** |
| EM-1 | No printer could be found |

### 

### 1.6 Class diagram and erd

#### 1.6.1 ERD

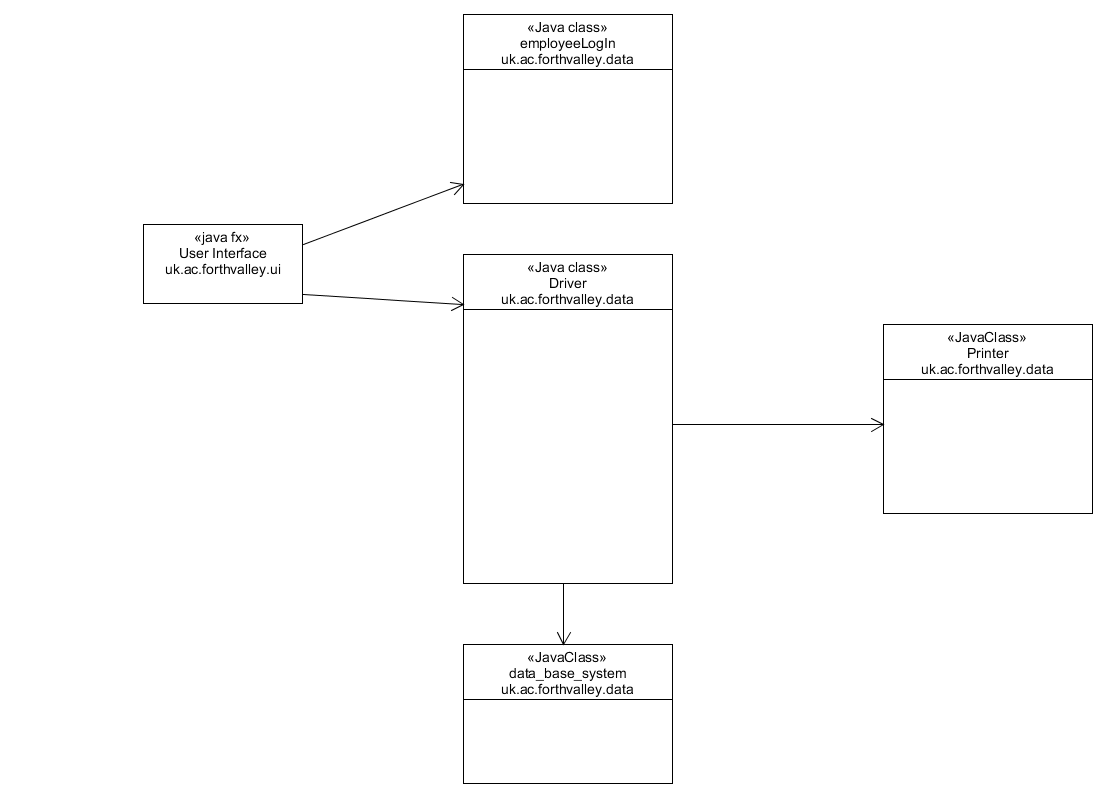
Following on from the crc cards that I made for the erd (section 1.3.1) I have created an initial erd that will be used as the storage method for the program currently the erd only shows the entities that will be in the program but will be expanded upon later to show the attributes and the data types of the attributes used in the erd.



### 

#### 1.6.2 Initial Class diagram

Initial class diagram for the application that shows what classes will be used in the program and how they interact with each other the way I made the diagram was taking some of the crc cards (section1.3.2) that I had created earlier to represent the classes that will be used in the program for now this diagram only shows what classes will be used in the program later in the design stage this will be elaborated on and will include the methods that will be used in the program.



## 

## View Model Analysis

### 1.7 Data binding model

The data binding model that I intend to use for the development of the application is that the front end of the application will have a connection to a phpMyAdmin data base so whenever the user interacts with the application there will be a data base in the back end of the program that will be accessed to retrieve any of the information that the user needs to use.

An example of this may be whenever a user requests to display information that request will retrieve the information from the database and display it to the user. the data base will also be able to save information that has been entered by the user.

During the design phase I will be going more in-depth with the data binding showing how my user interface and data base will work together by showing what happens to the data from the user interface and showing how data from the data base is displayed to the user interface after it has been retrieved from the data base. I will also show how information entered by the user accesses the data base and what happens to that data when it is being stored.

### 1.8 UI design

Before starting the creation of the user interface, I performed some research on different websites on some of the most important features of good ui design to help me decided how I was going to design the user interface after performing this research on different design techniques I came to the conclusion of how I am going to design the user interface.

I am going to design a simple to navigate interface that displays information clearly to the user I am not going to use many bright colours that distract the user when using the system Another thing that I am going to keep in find during the creation of the user interface is to make sure all my pages have a clean consistent layout that all follow the same colour scheme. The user interface will be designed in a way that all the information that the user needs to find is clearly displayed to them.

The way text will be displayed in the application is to use be using colours with high contrast with the background the text is displayed in for example If the back ground uses light colours, I will be using black text and if the background uses dark colours I will be using white text the reason that I am doing this is so information the application displays is easy to read for a user.

Another way that I will be making the application easy to read is to use a large text size around size 12 rather than using small font I will also a font that easily stands out to make the information easy to read.

The way that I will be designing the user interface is to intital build some basic wire frames which show the lay out of the application then pick a few different colour schemes and take them to the client and allow them to pick the one they like the most after letting the client decided which one they like the most I will then add that colour scheme into the wire frames that I created.

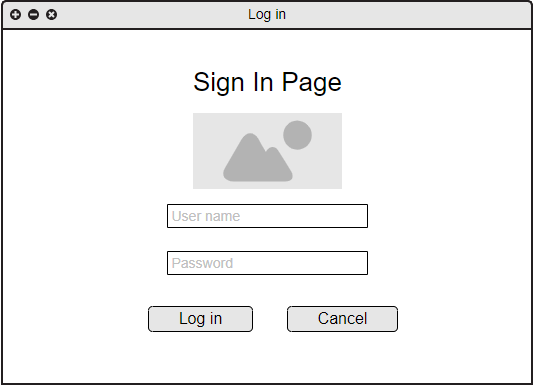
After designing the wire frame the way I will be building the user interface is using java fx.

#### 1.8.1 Initial Wireframes

I have created some initial wires frames that may be iterated up on in the design phase the wireframes that I have created help me get an idea of what the application could potentially look like.

I will be showing the initial design to the client to get feedback on what they like and dislike about the interface in a later version I will also be adding the chosen colour scheme to the wire frames.

**Log in page**

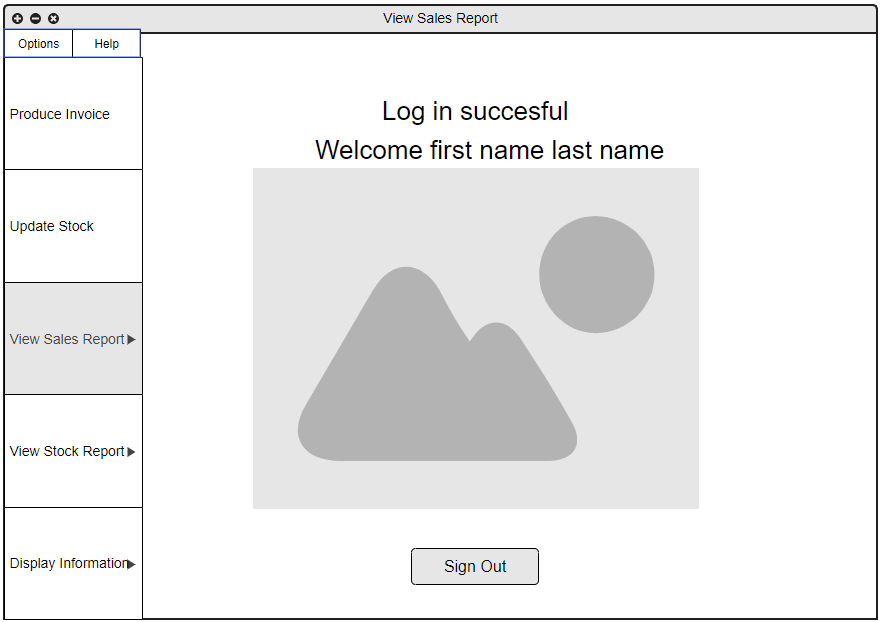


This wire frame shows what the log in screen for the application will look it is the first page that the user will see when the application is loaded up the user will be able to enter their user information and click the log in button and the cancel button which will close the application. When the user logs in if they give invalid details, they will get an error message that tells them they have entered invalid information this error is shown in the next diagram.

**Log in page invalid information**

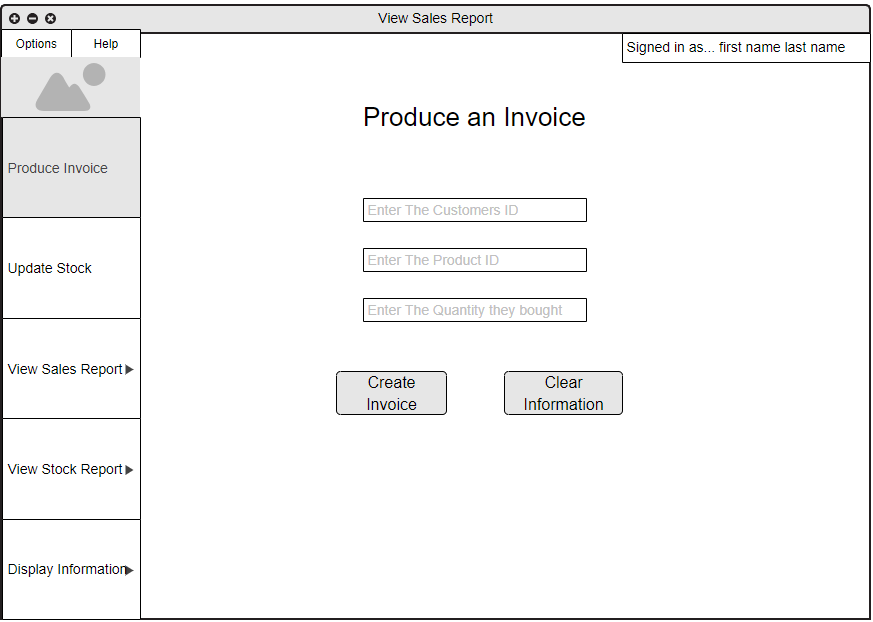


**Welcome screen**



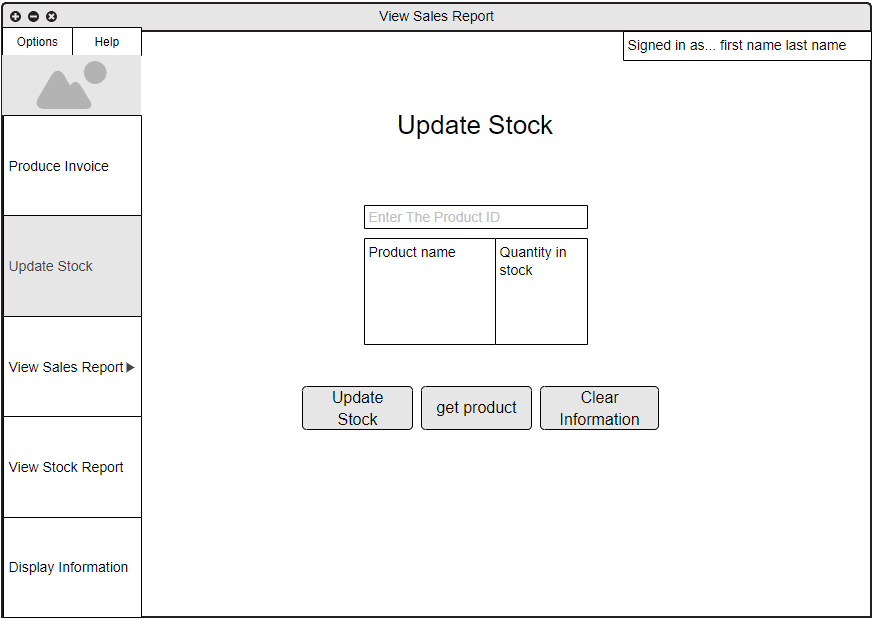
After a user logs in successfully they will be taken to the home screen of the application this page will display who is logged in and allow the user to navigate to which section of the application they want to use this page also has a sign out button which will sign the user out and take them back to the log in screen.

**Produce invoice screen**

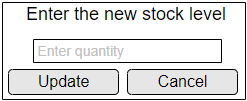


This wire frame shows the screen that will create the invoices the user will be able to enter the information they want for the invoice by entering the ids of the customer, product and quantity purchased the user will then be able to press the create invoice button that will then create the invoice and store it in the data base for future use. The user will also be able to press the clear information button that will remove anything that is currently in any of the text fields.

**Update stock**

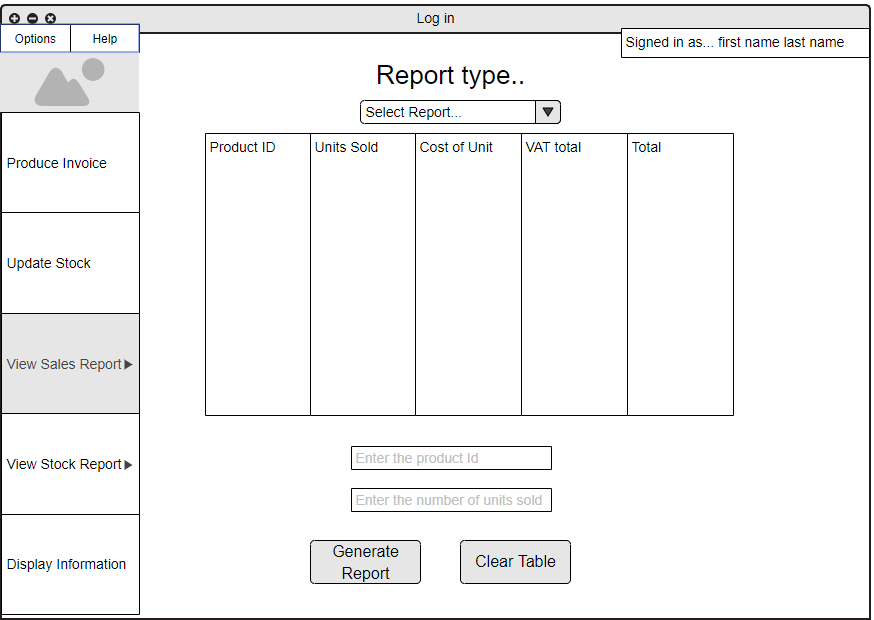


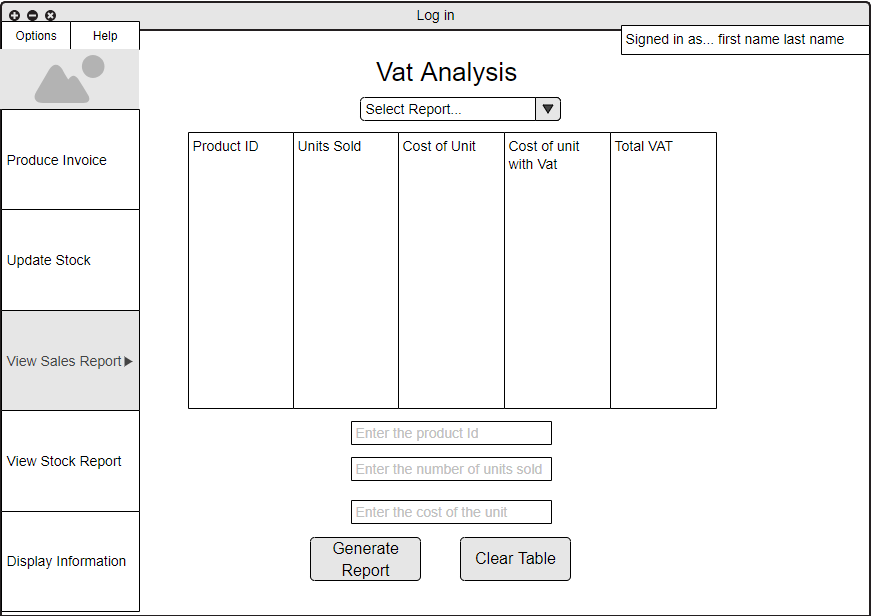
This wire frame shows the update stock interface the user will be able to enter a product id which will then be displayed in table it shows the name of the product and how much stock there is currently when the user presses the update stock button, they will be shown this screen

This allows the user to enter the new stock of the item after the user presses the update button the change will be made to the system and the new stock value will be shown to the user the user will also be able to press the cancel button if they decided that they no longer want to update the value.

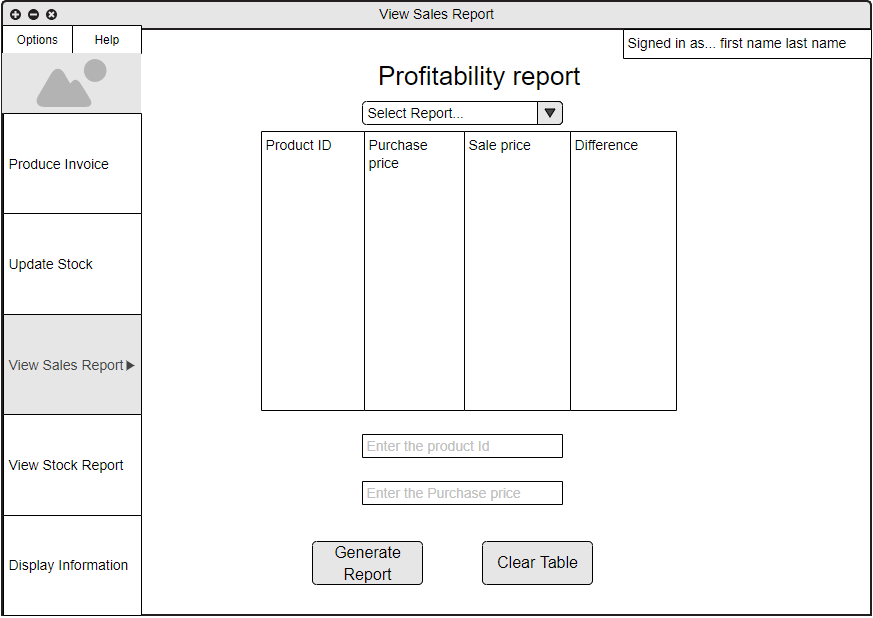
**Generate reports**

**Generate report – financial wireframe**

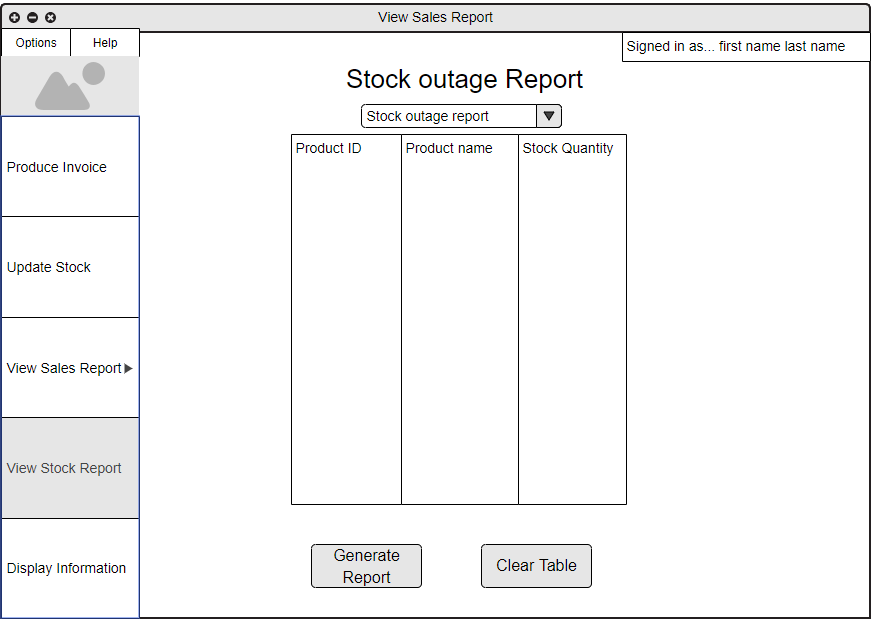


**Generate report - vat analysis wireframe** 

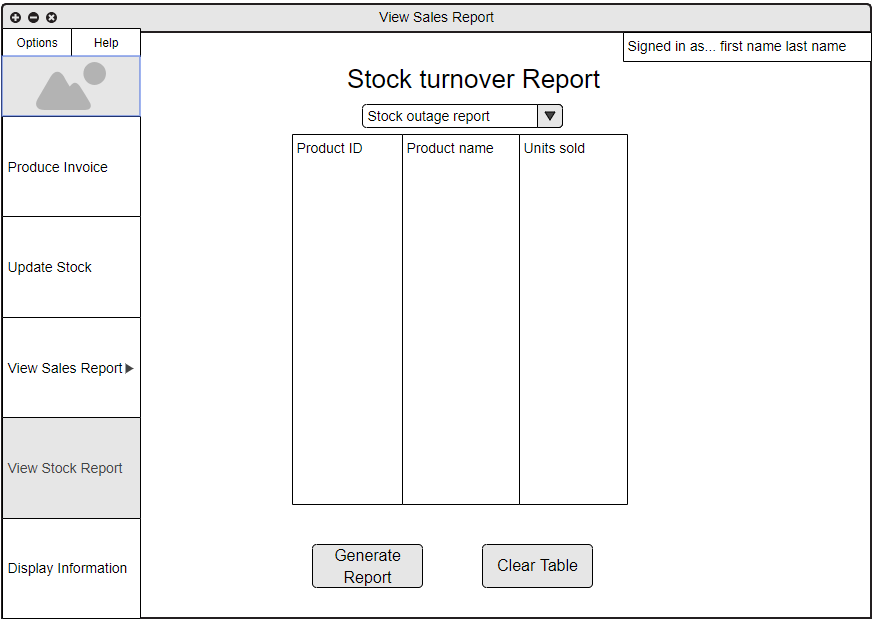
**Generate Report – profitability report wireframe**



**Generate report – stock outage wireframe**



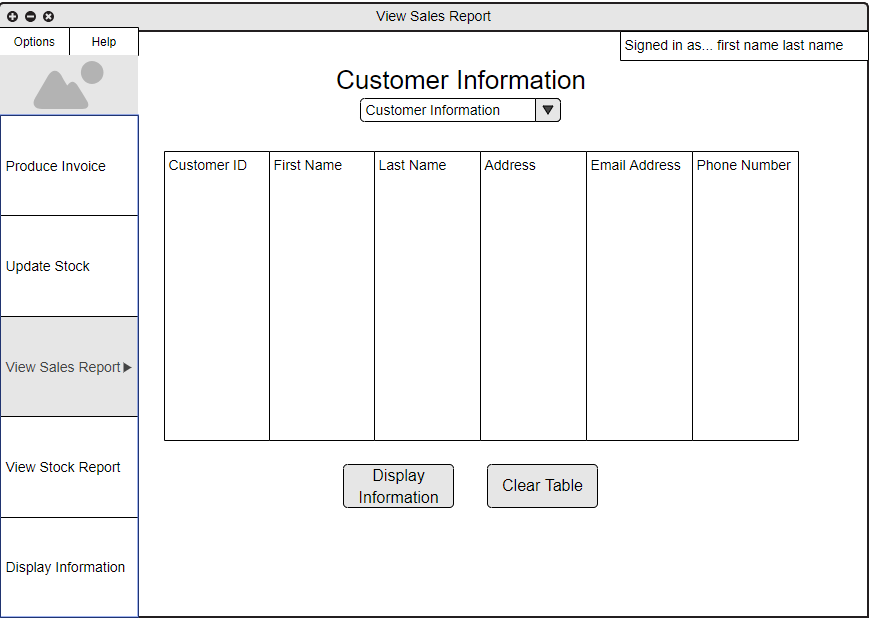
**Generate report - stock turnover report wireframe**



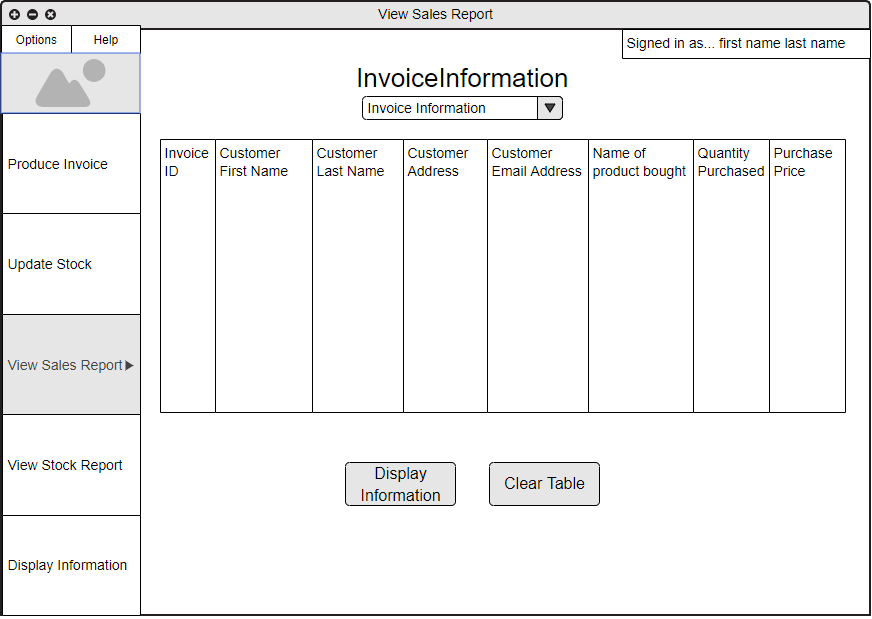
These frames show the wire frames for the different types of the reports that users will be able to generate the functionally of these reports is similar for all of them the user will enter the required information to generate the report then will press the generate report button the table will then be populated with the required information to be displayed to the user. The user will have the option to clear the table as well.

**Wire frames information display**

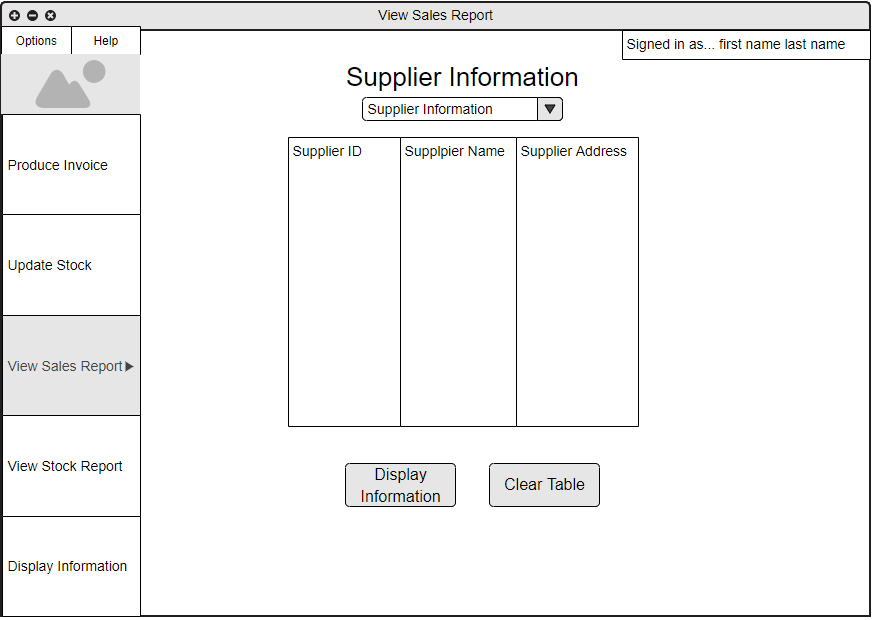
**Display customer information wireframe**



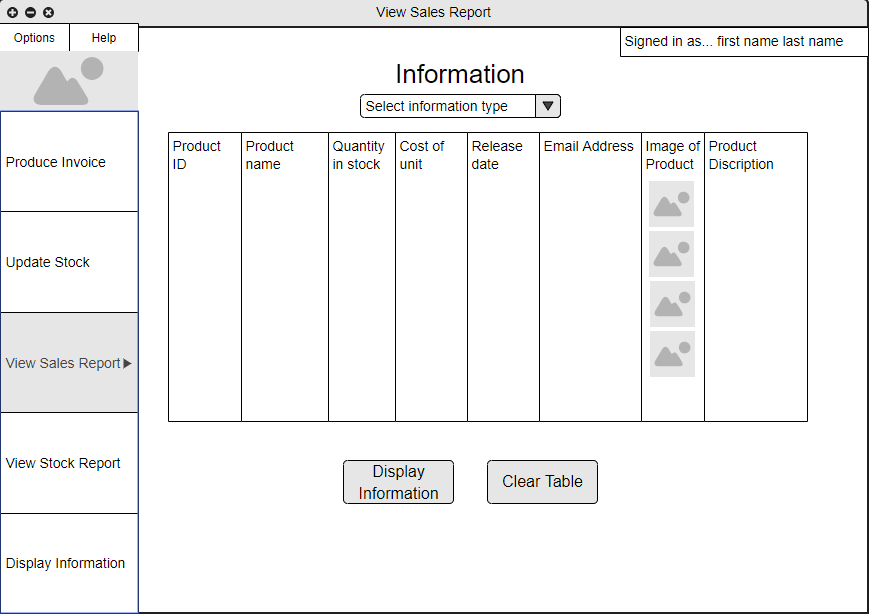
**Display invoice information wireframe**



**Display supplier information wireframe**

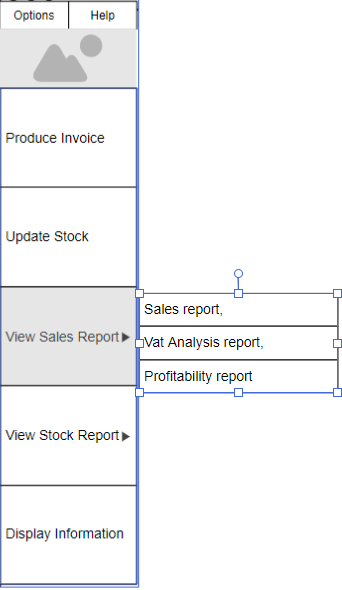


**Display product information wireframe**



These wire frames show the information that will be displayed once the user selects what type of information they want to view is the way these will function is that the user will be able to select the type of information they want to view from the drop down menu the user will then press the display information button which will create a list of all the information they have stored about what they want to view.

#### 1.8.2 Application navigation

The way the application is navigated is by having a simple menu system that allows you to click on what page you want to be displayed on the parts that have an arrow indicated that there is more than one page that is relevant to that section of the application and when hovered over additional pages may be opened (refer to example of hover over on menu).

The user will also be able to press the image to take them back to the welcome screen of the application.

When the user selects the page, they want to view that pages navigation icon will change colour to clearly distinguish what page they are currently on.

When the user clicks on the option, they want the page that they are on will change to the relevant page and the navigation menu will indicate to the user what page they are on by highlighting the page they selected.

There is also an option and help menu the options menu will have a drop-down box that will have the options for the user to exit the program and a log out button and the help button will display information on how the application is used.

The reason that I decided to use this kind of menu for the navigation of the application was I feel this type of design keeps the application simple to navigate while also still looking visually appealing.

**Navigation map**

To show how the application is navigated I have created a navigation menu this diagram shows the ways that the application can be navigated each of the boxes in the diagram represent a page that can be viewed. To indicated menus that are hover over and have more than one page that can be viewed I have added extensions.

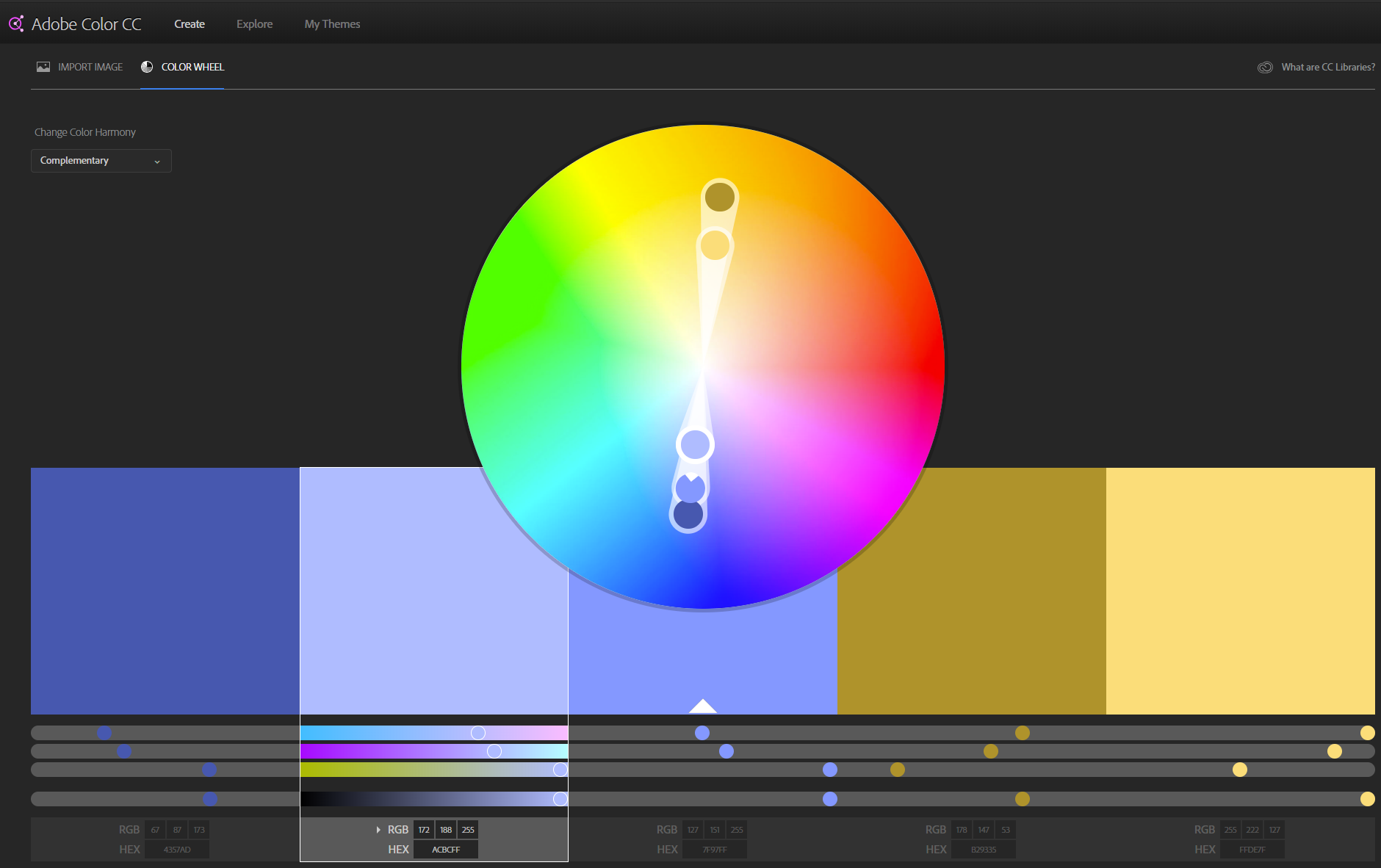
### 

#### 1.8.3 Colour scheme selection

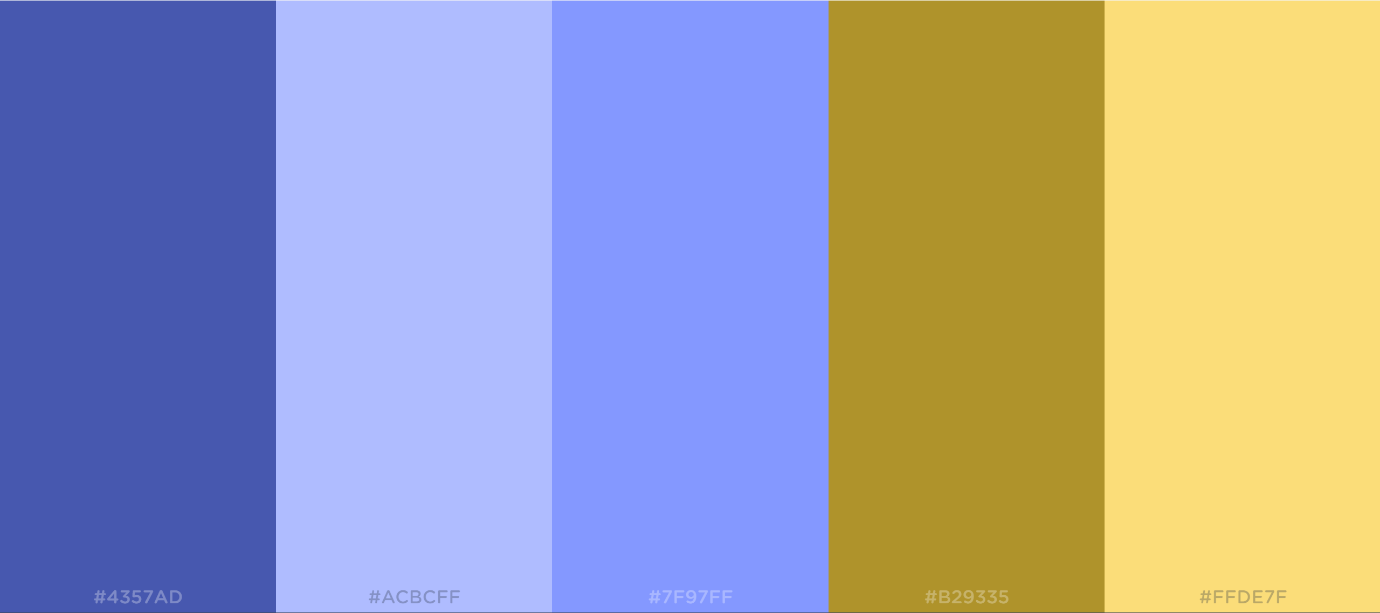
The first thing that I done was to perform some research on what on the different types of colour blindness to get idea of how the colour scheme should be designed and to avoid any colour scheme combos that could not be used the people who suffer from colour blindness. After performing my research keeping in mind what I found out I used adobe colour picker to select colours that work with each of to come up with my colour scheme after selecting the colours that I wanted to use I used an application that would transform the colours that I selected into what people with different types of colour blindness would see. After finding a colour scheme that would work, I then added the scheme on to a few pages of the application.

After doing this for three different colour scheme I will be showing the client each of the choices and giving them a questionnaire to fill out to allow them to pick what one the like the most or if they find anything, they don’t like about the choices.

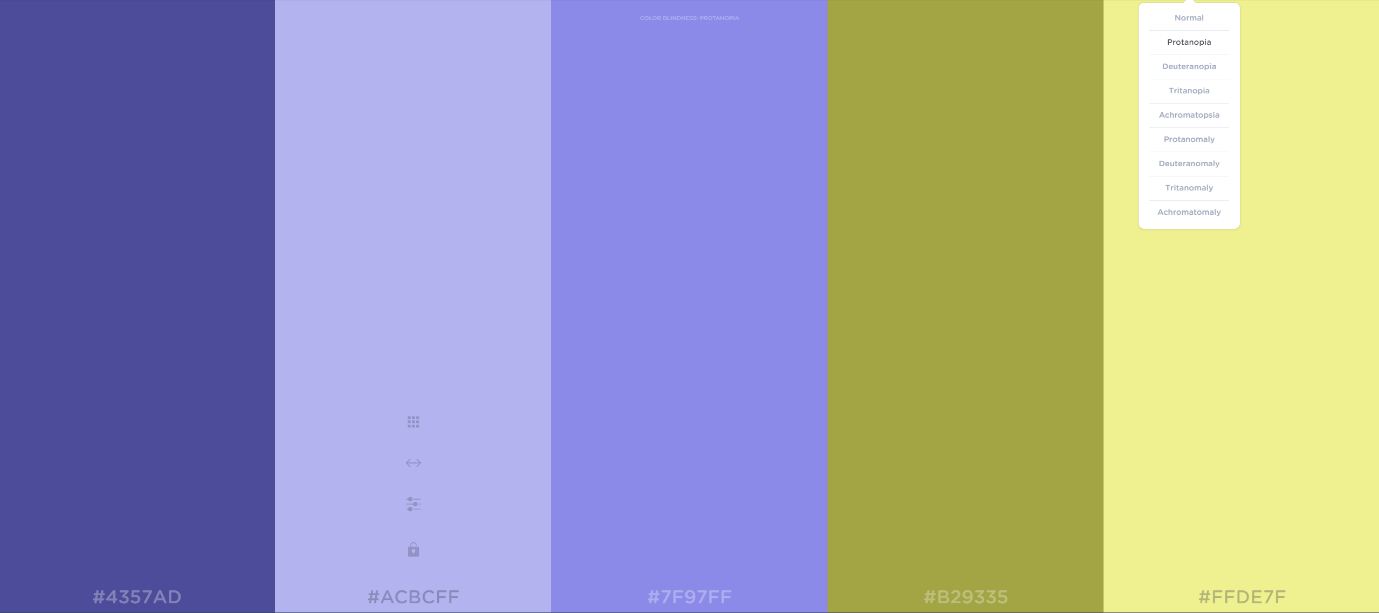
#### Colour scheme choice 1



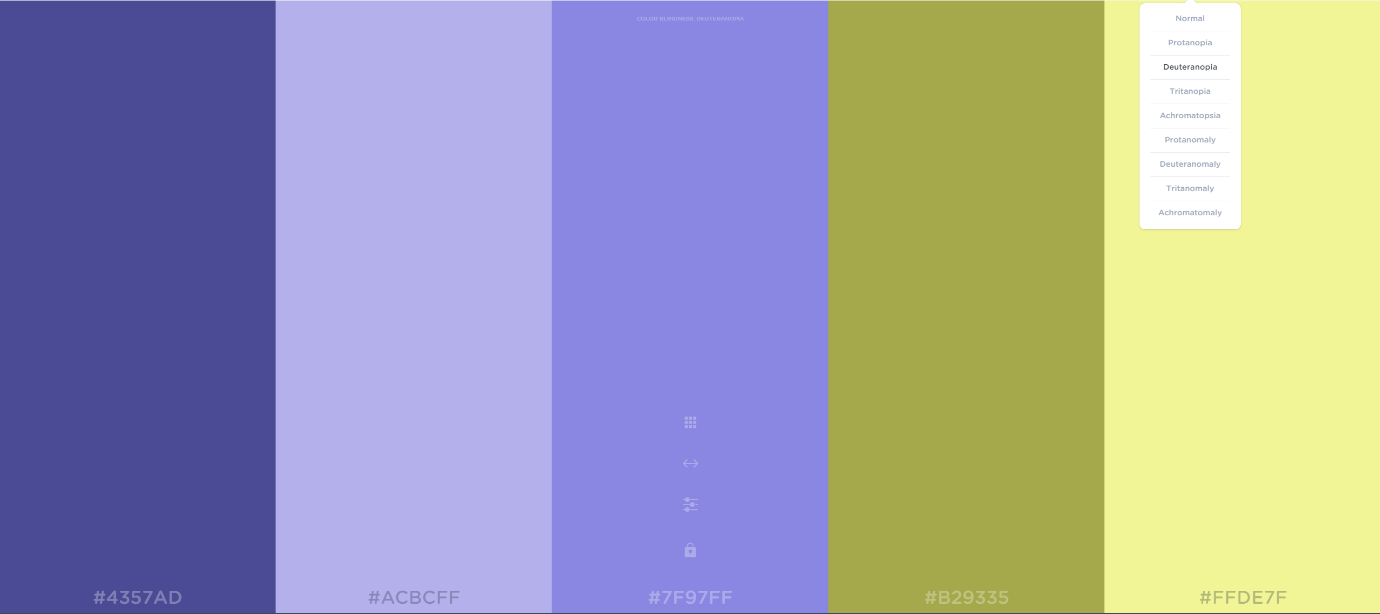
normal



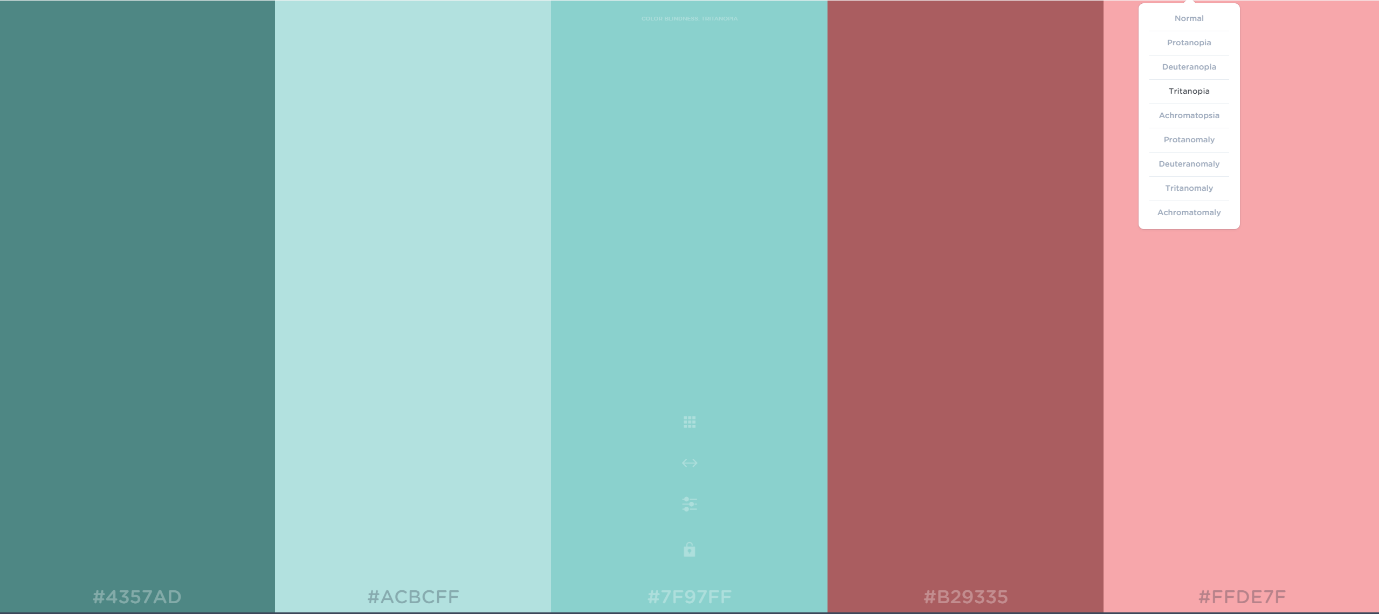
Protanopia



Deuteranopia



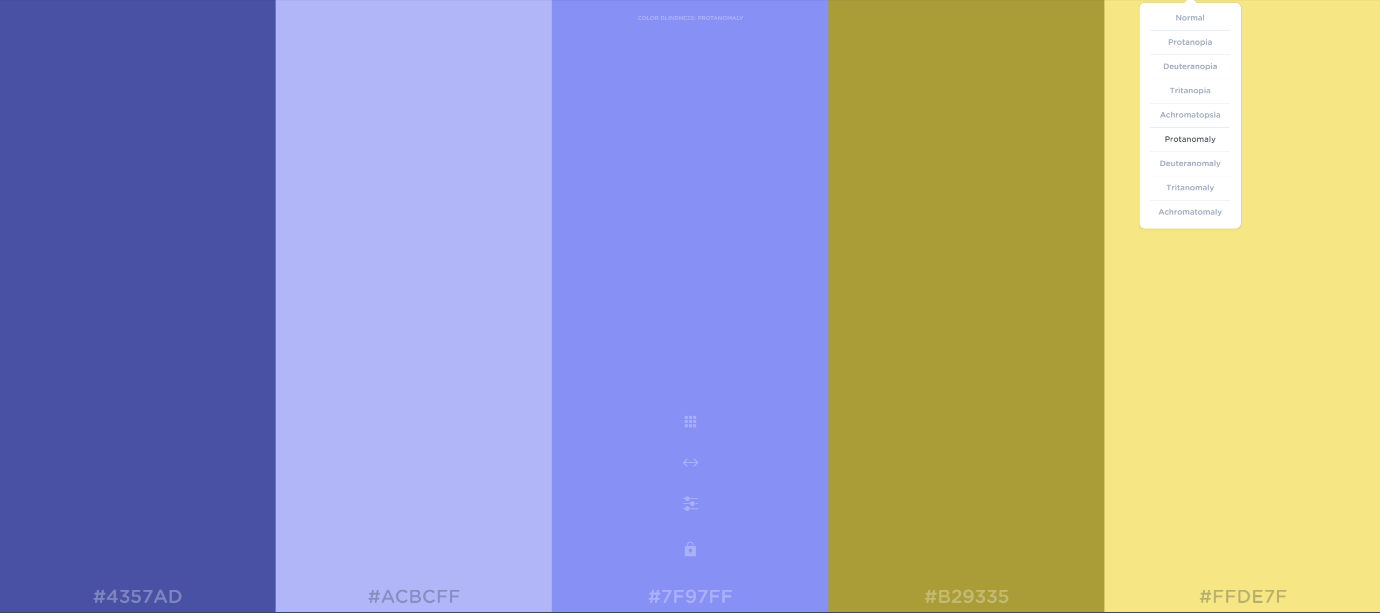
Tritanopia



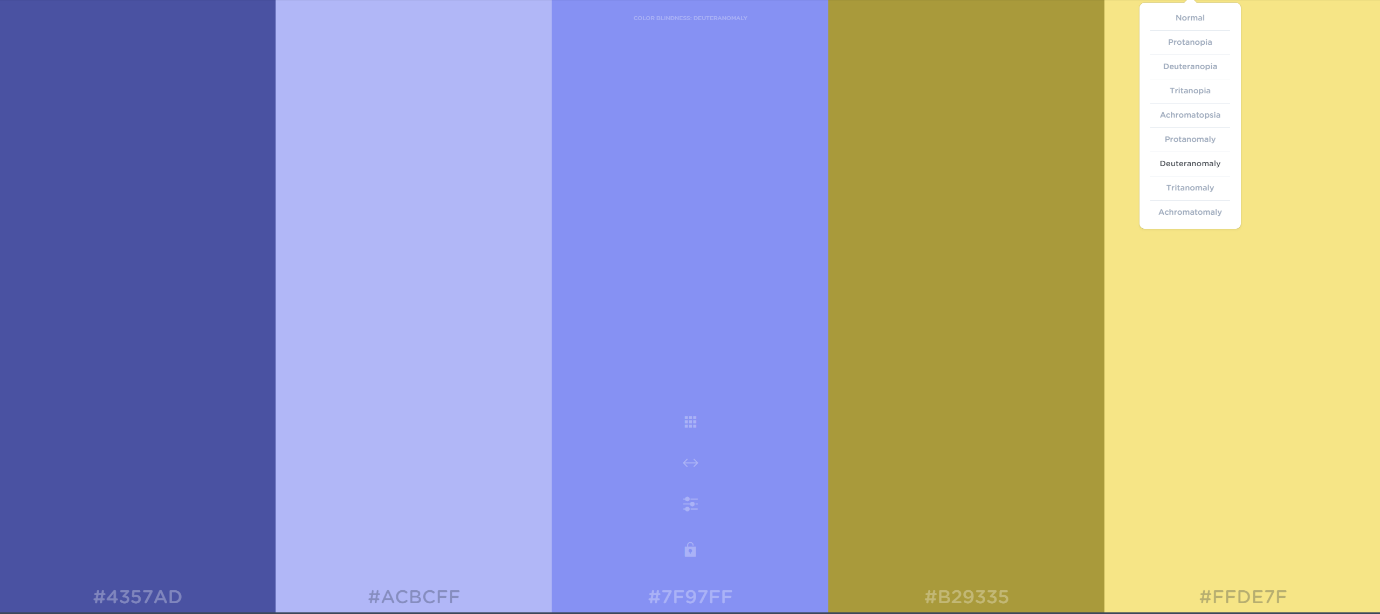
Achromatopsia



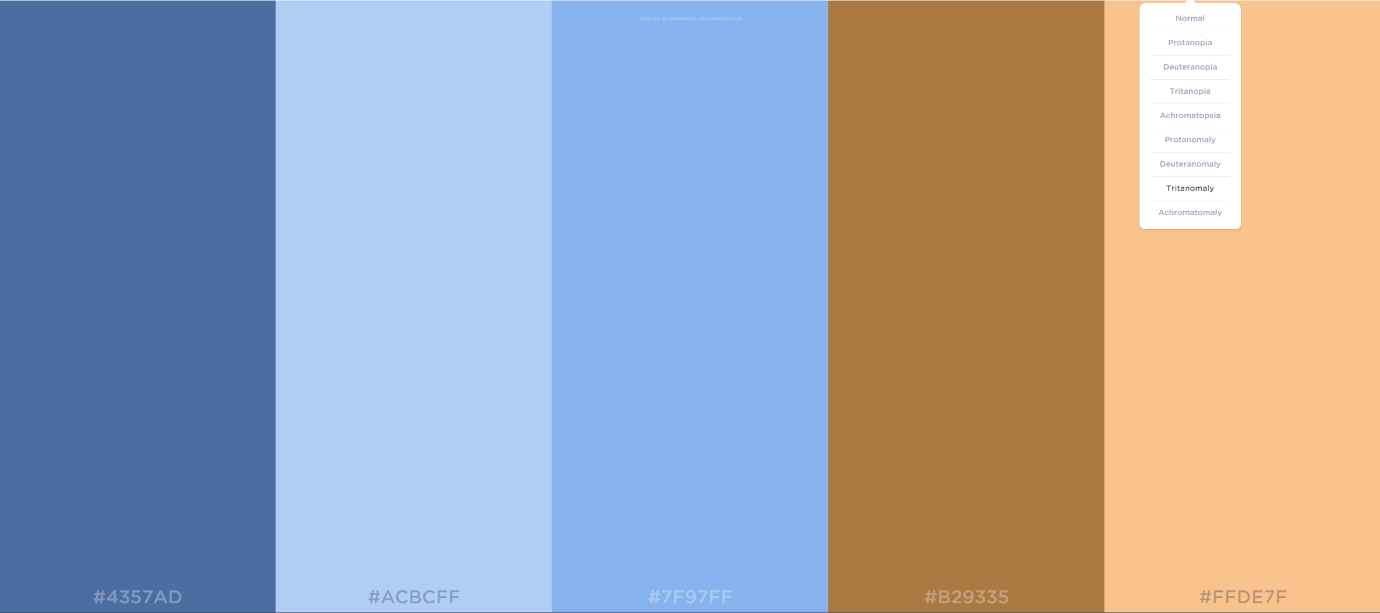
Protanomaly



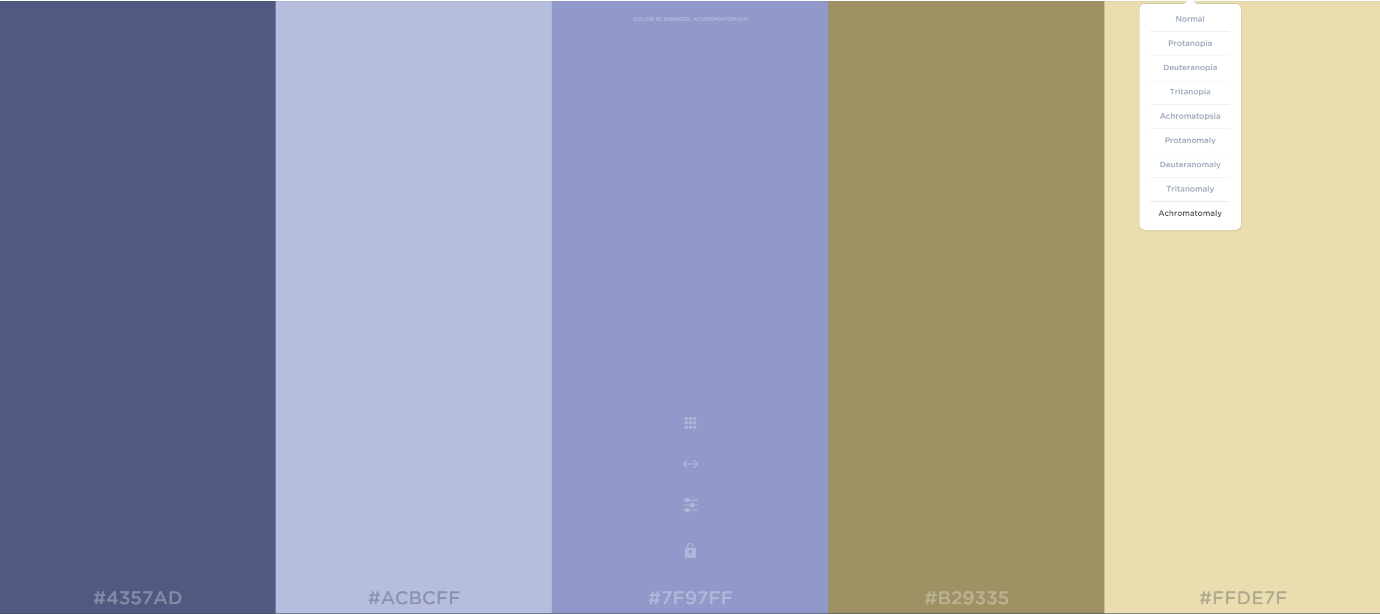
Deuteranomaly



Tritanomaly

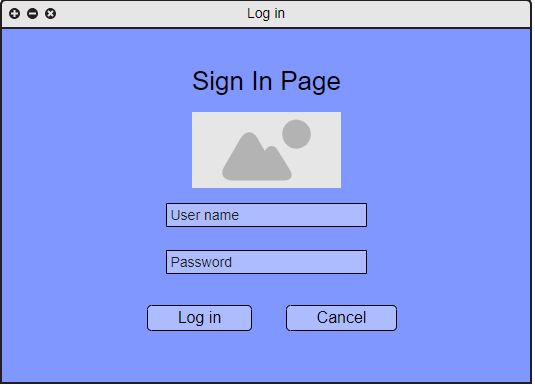


Achromatically



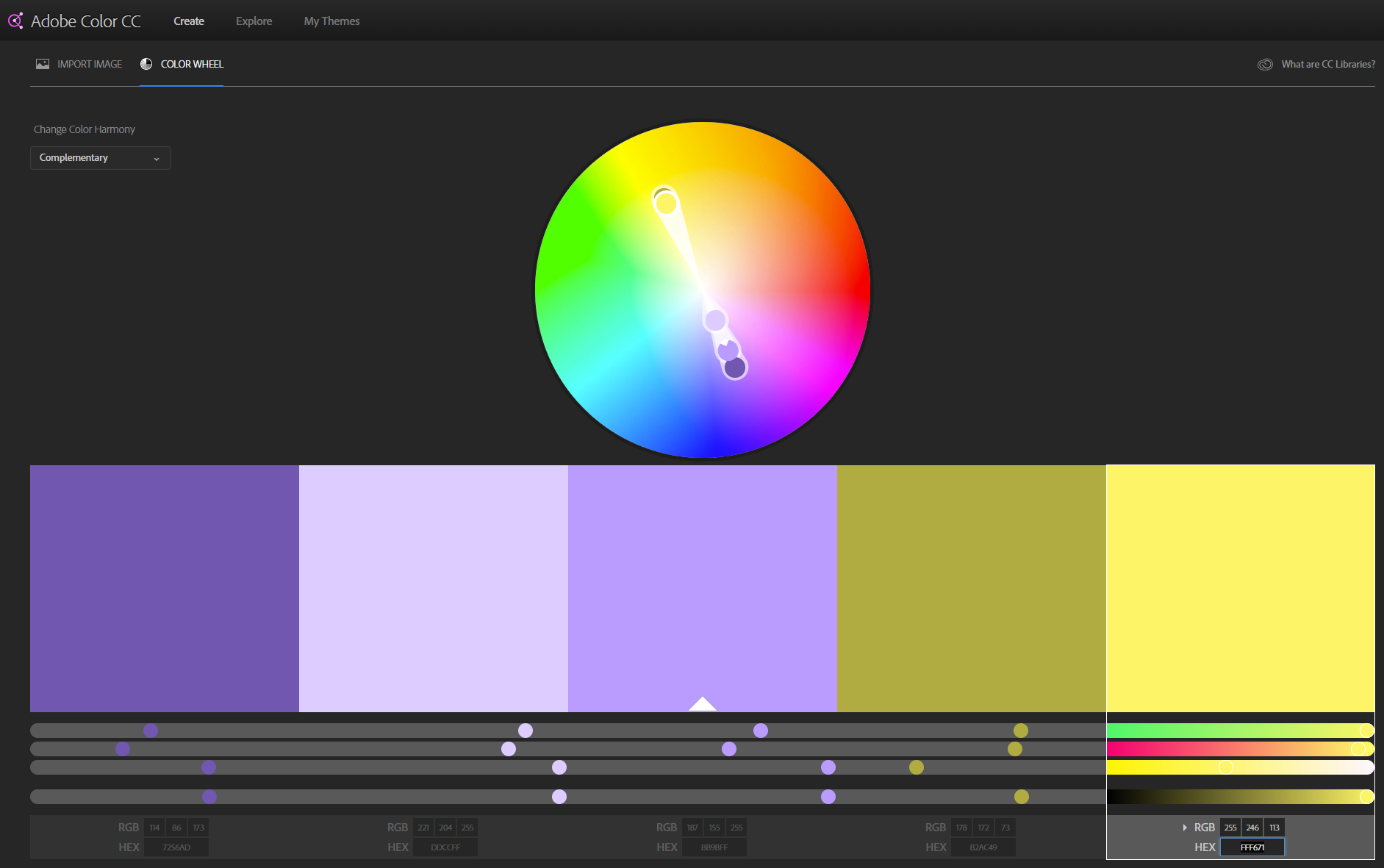
Application using choice 1

##### Login page colour scheme 1

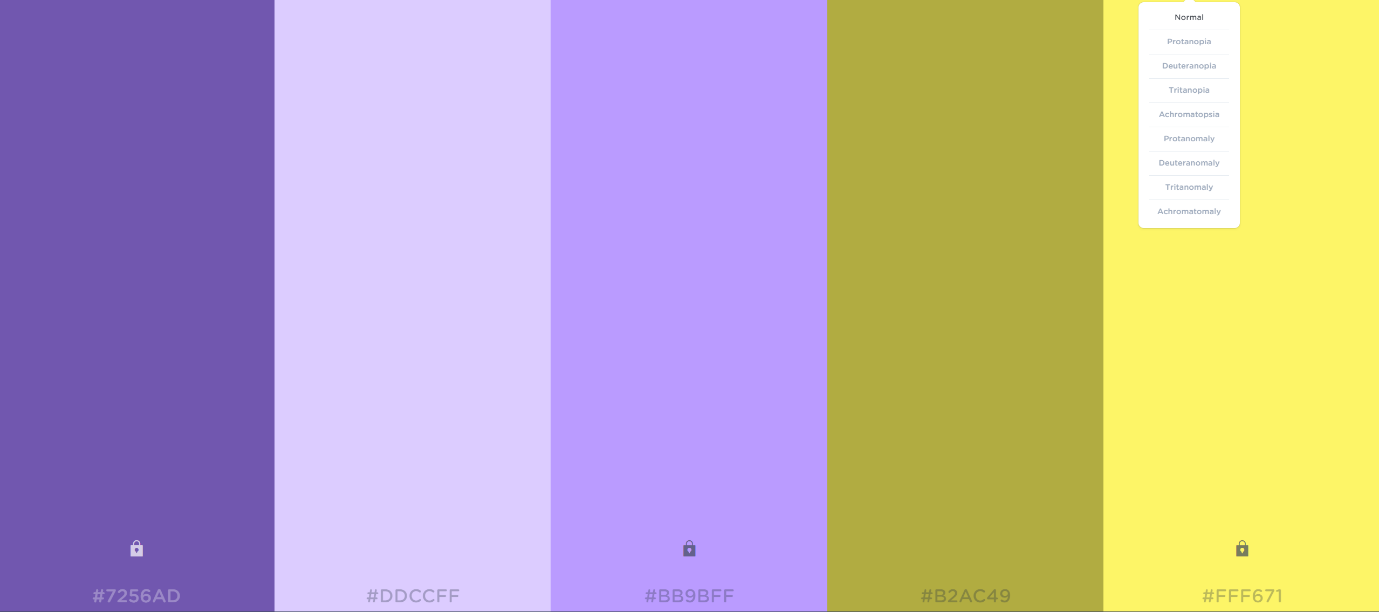




Colour scheme 2



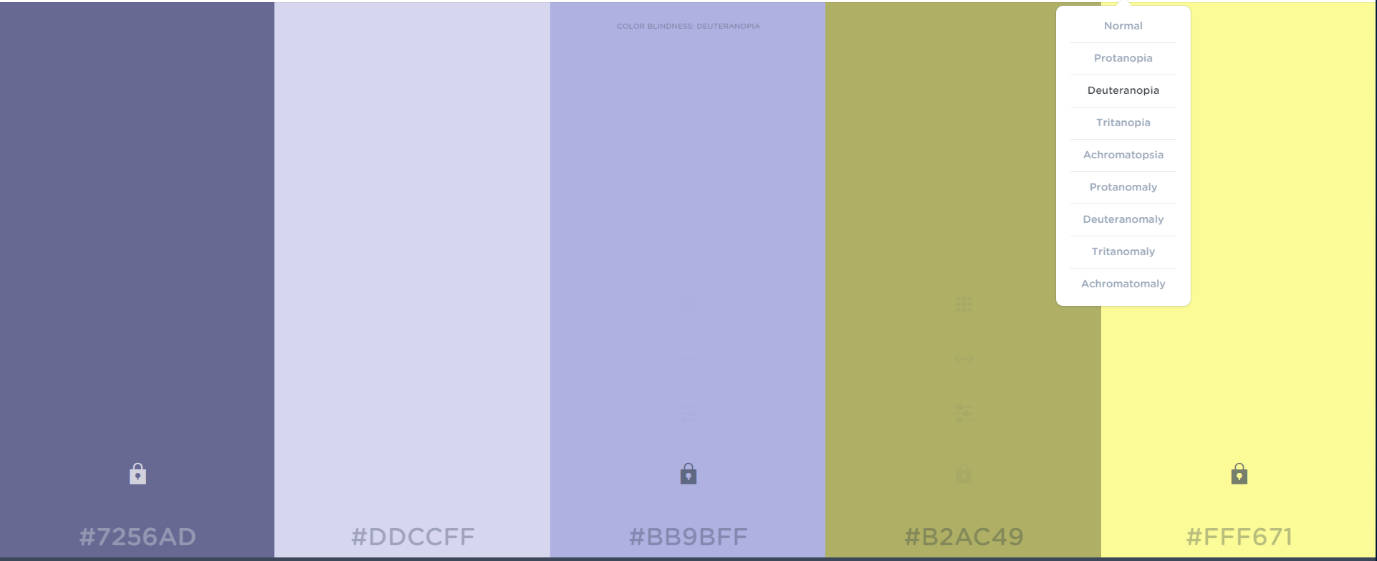
Normal



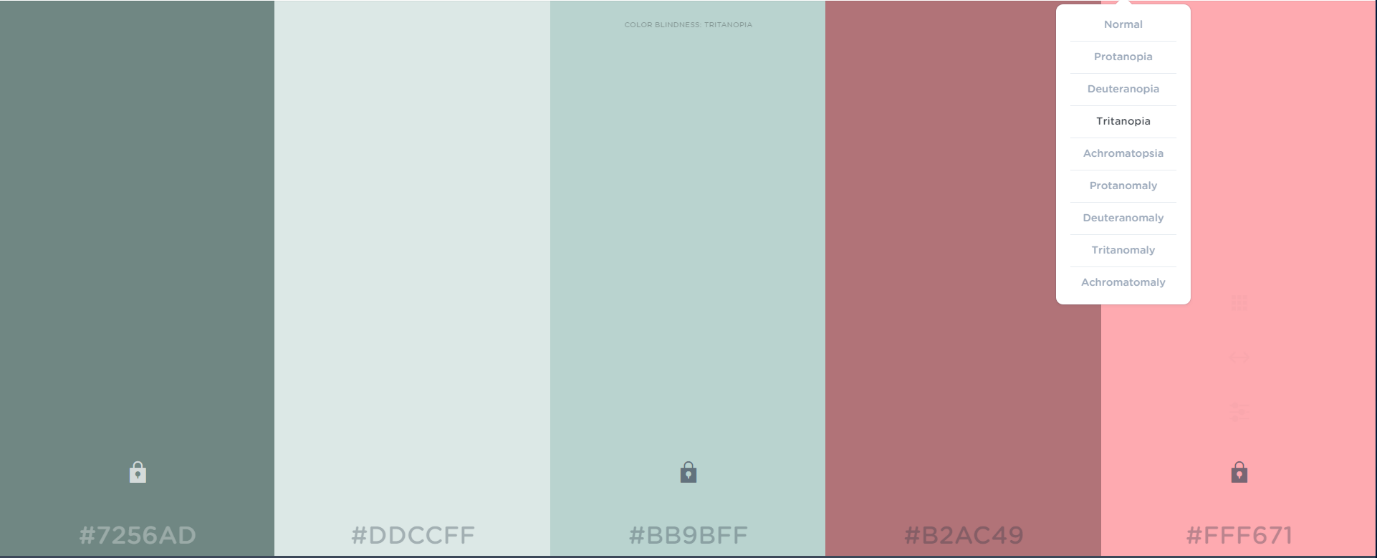
Protanopia



Deuteranopia



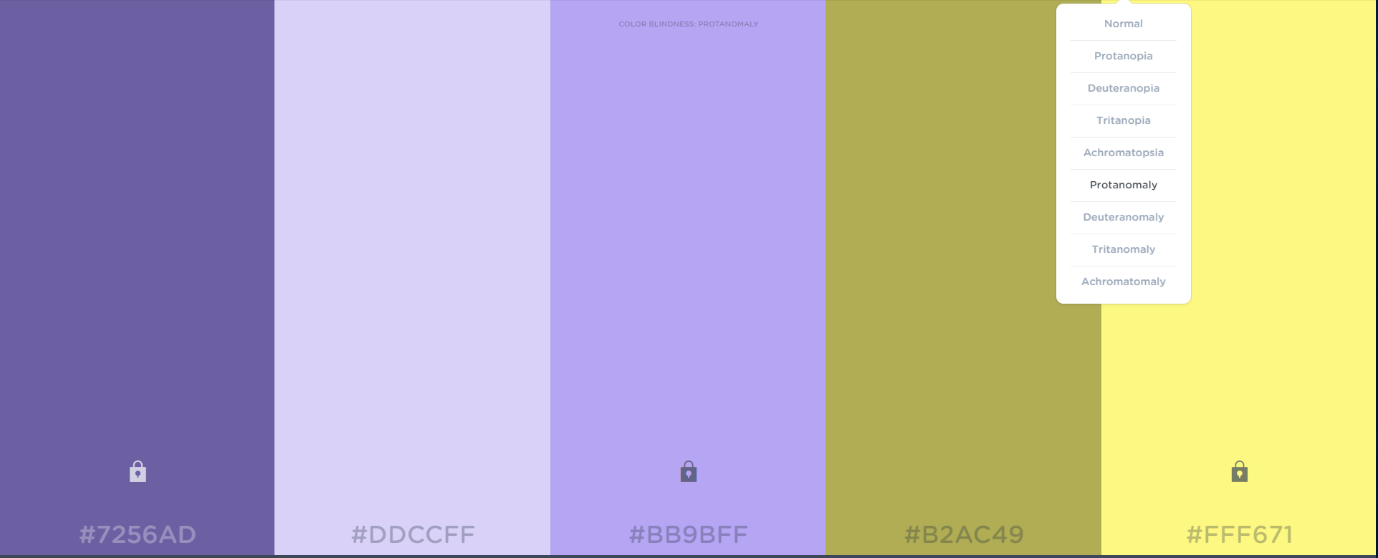
Tritanopia



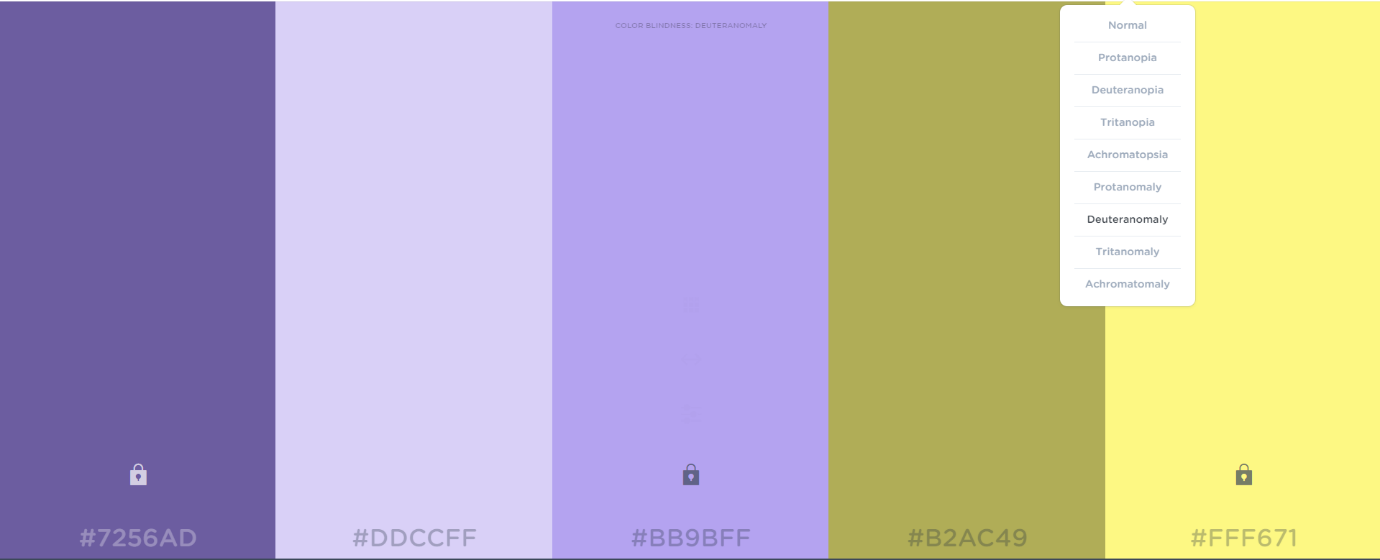
Achromatopsia



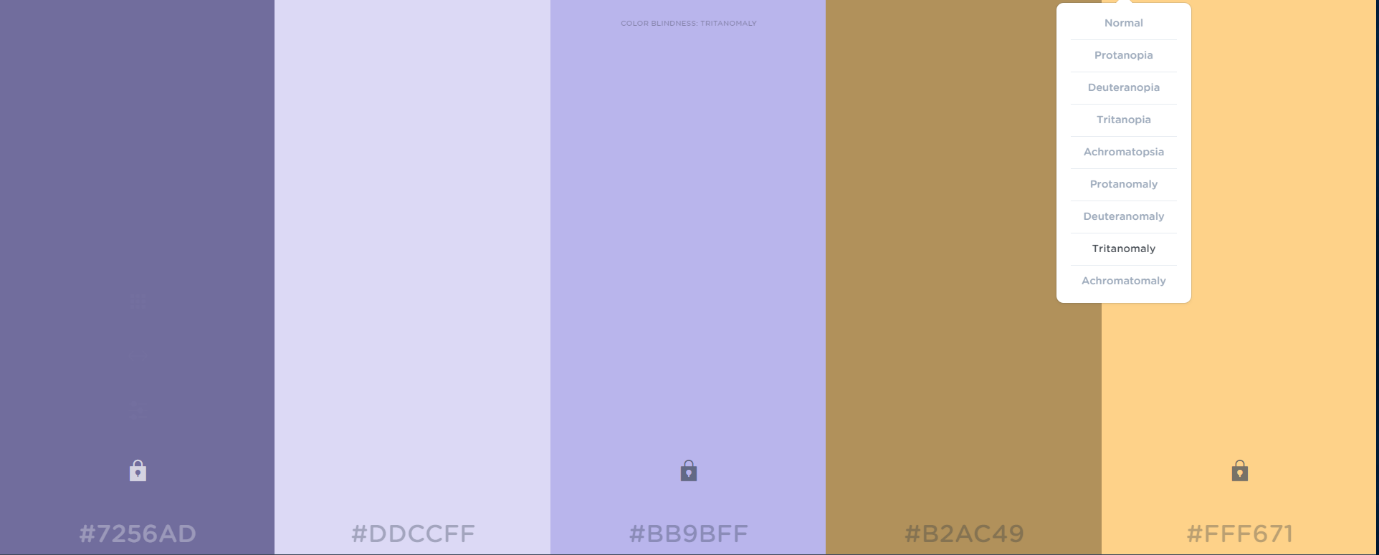
Protanomaly



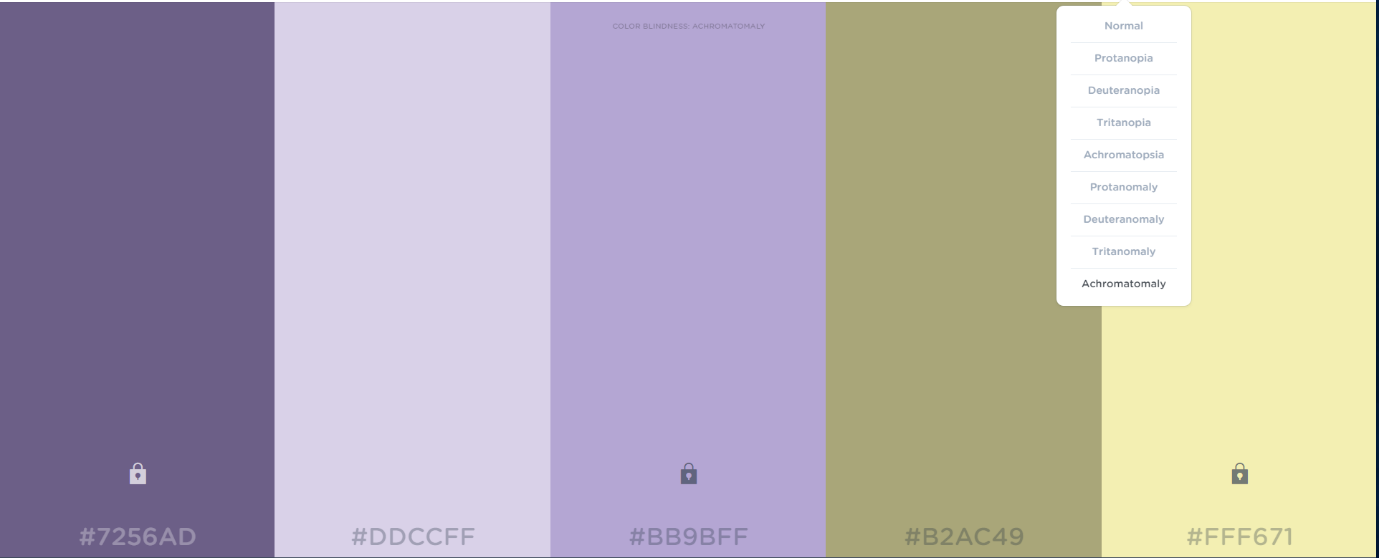
Deuteranomaly



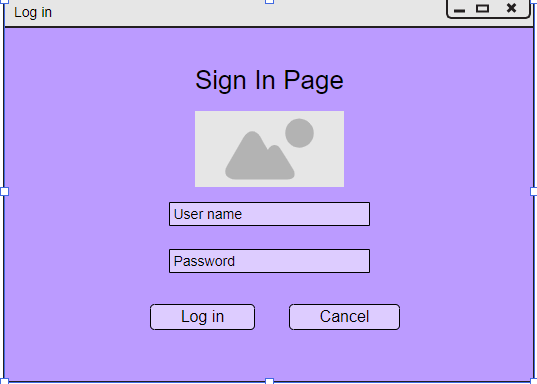
Tritanomaly

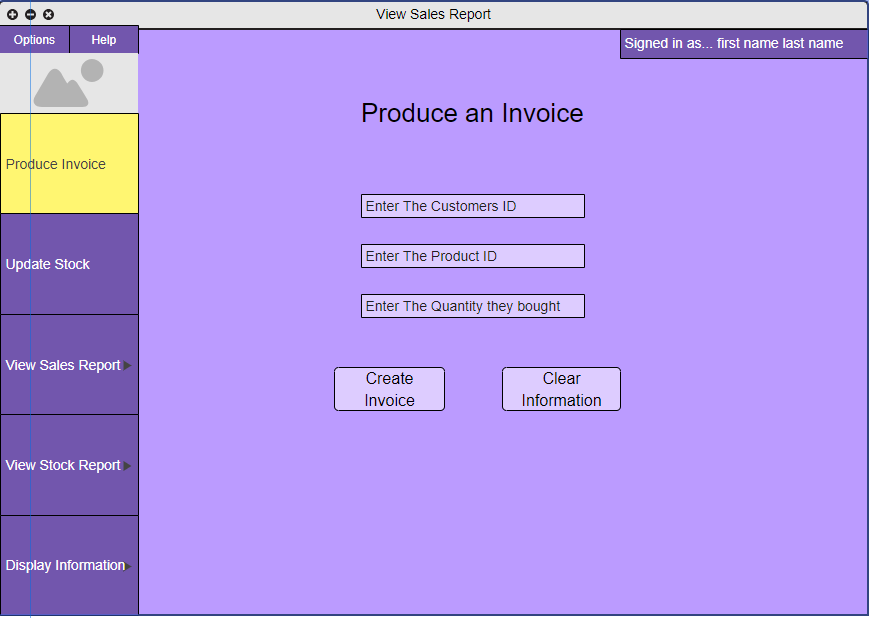


Achromatoamly



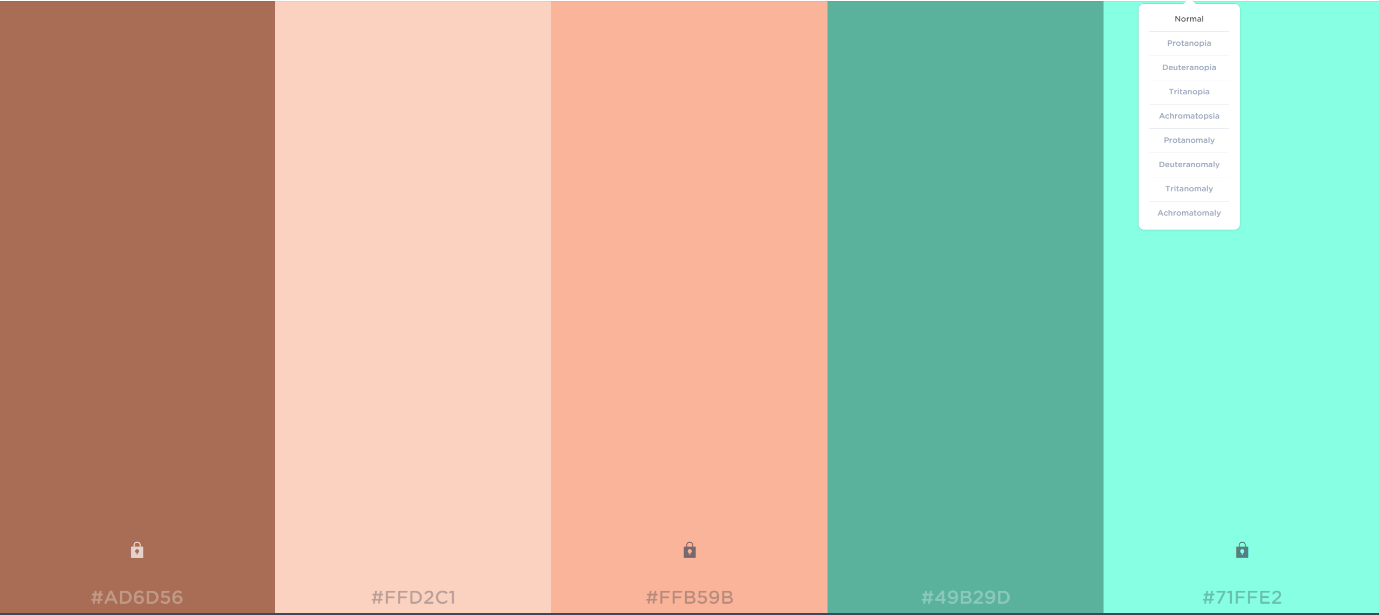
**Log in page colour scheme 2**



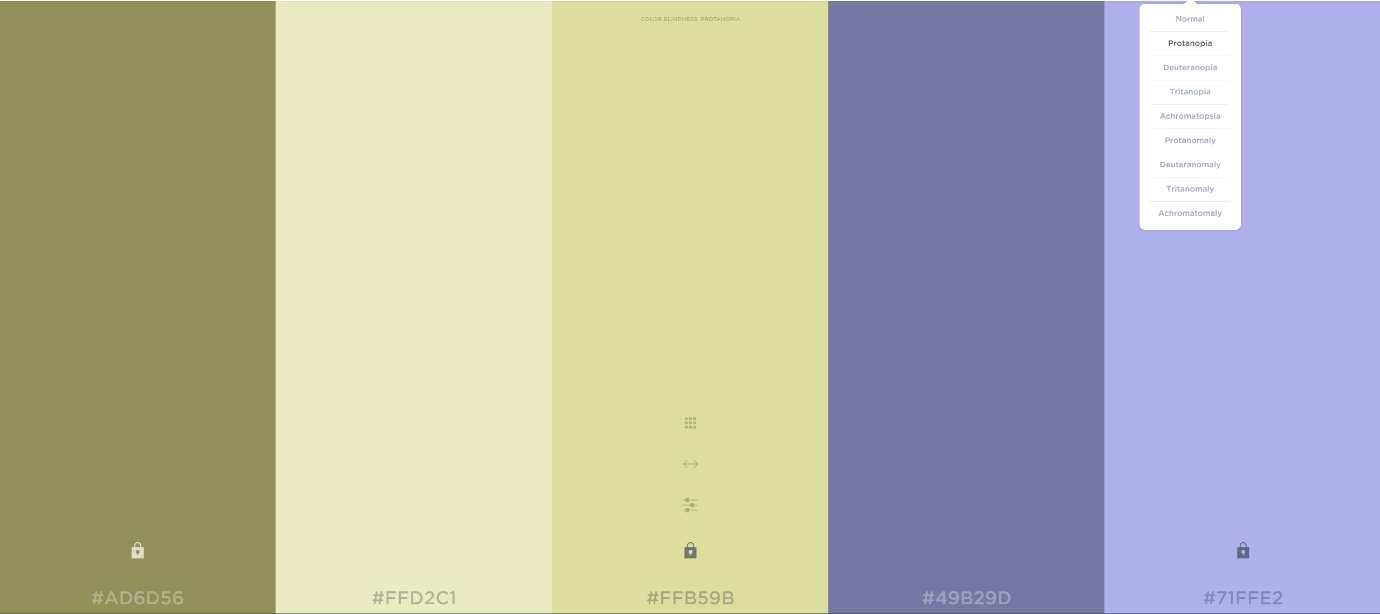


##### 1.8.3.3 Colour scheme 3

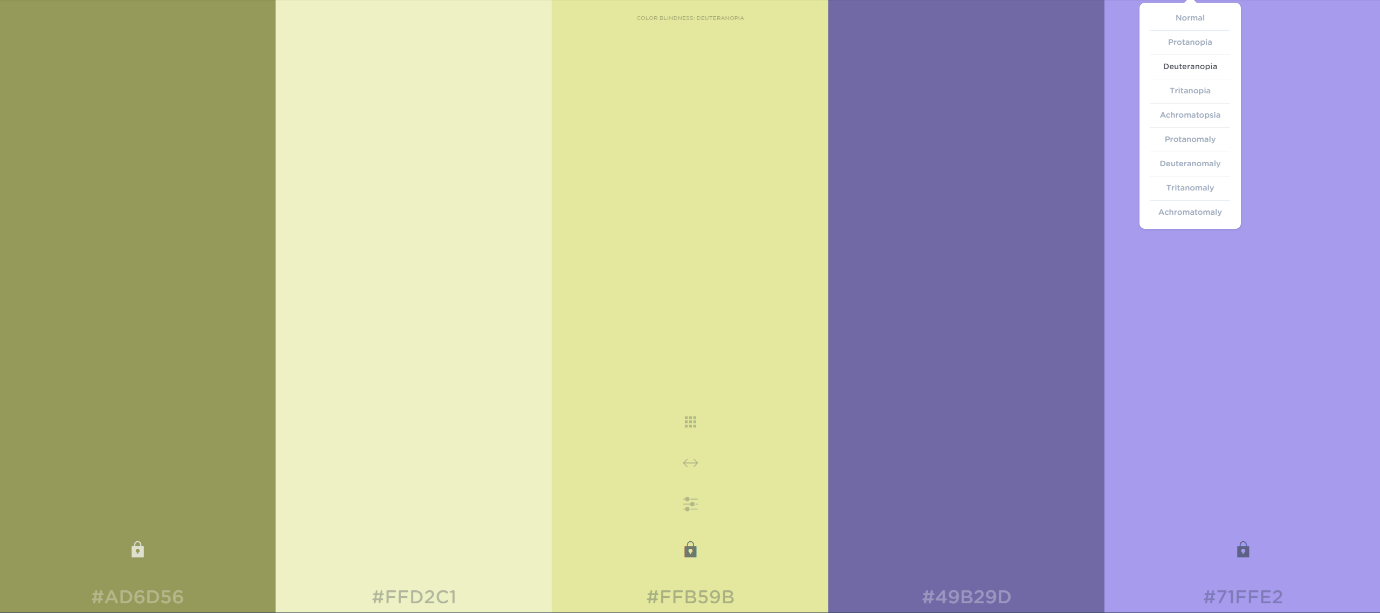
Normal



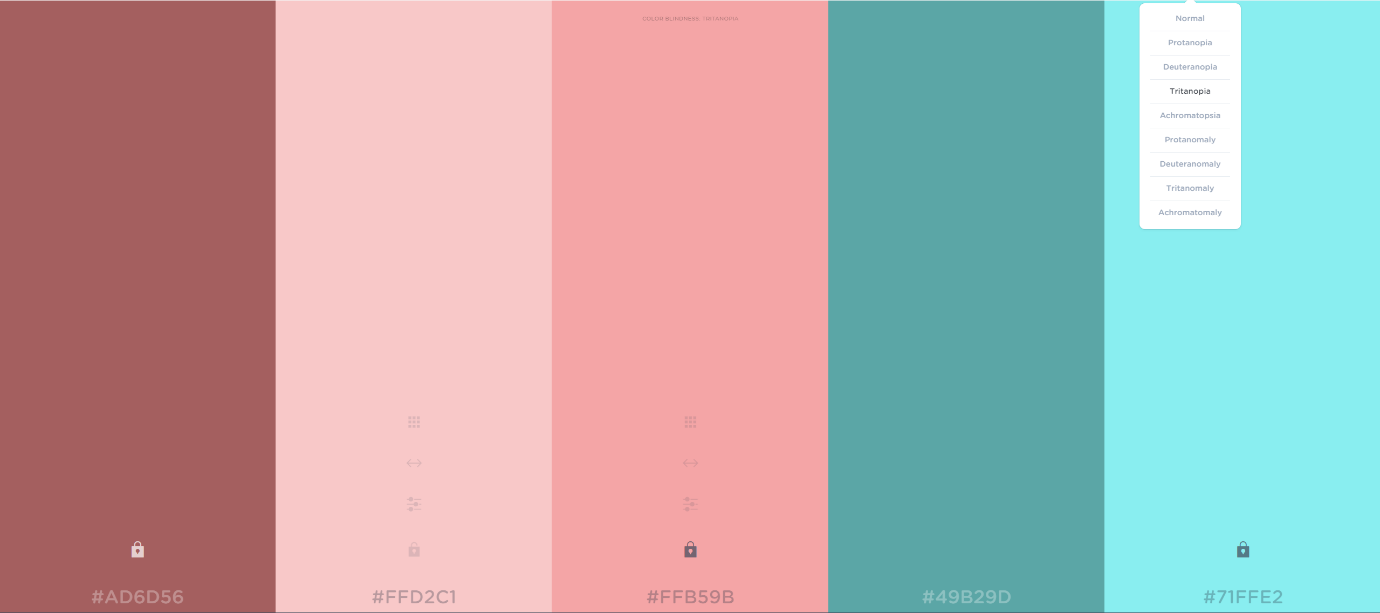
Protanopia



Deuteranopia



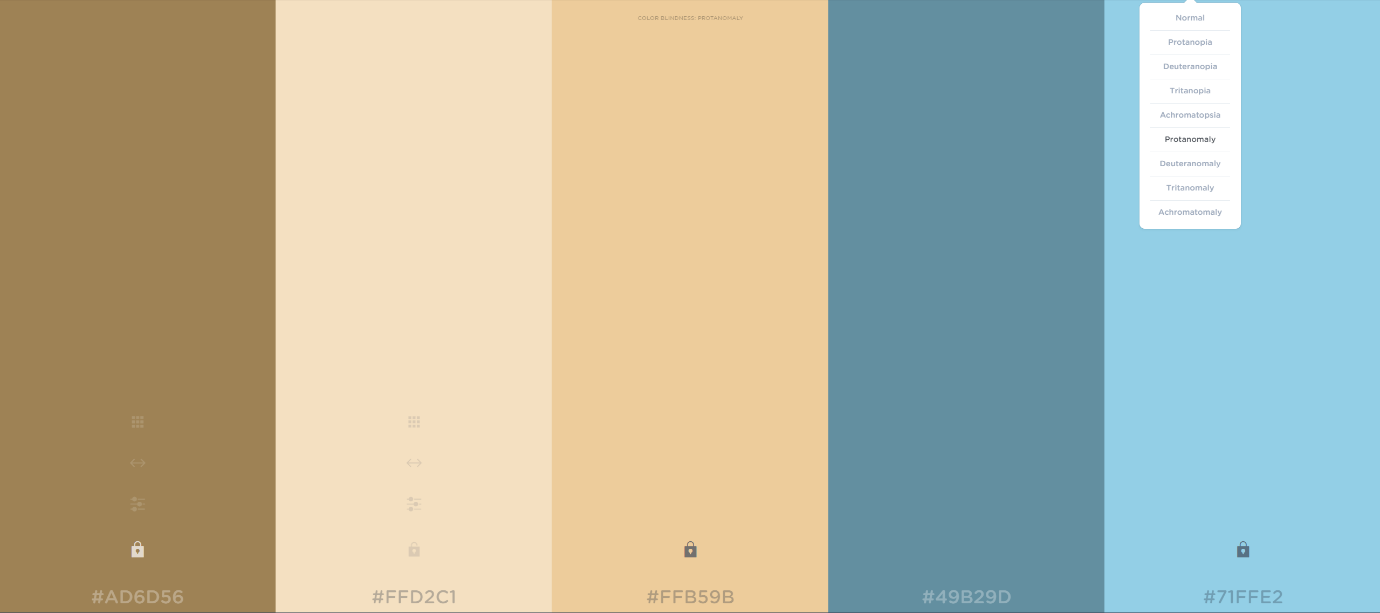
Tritanopia



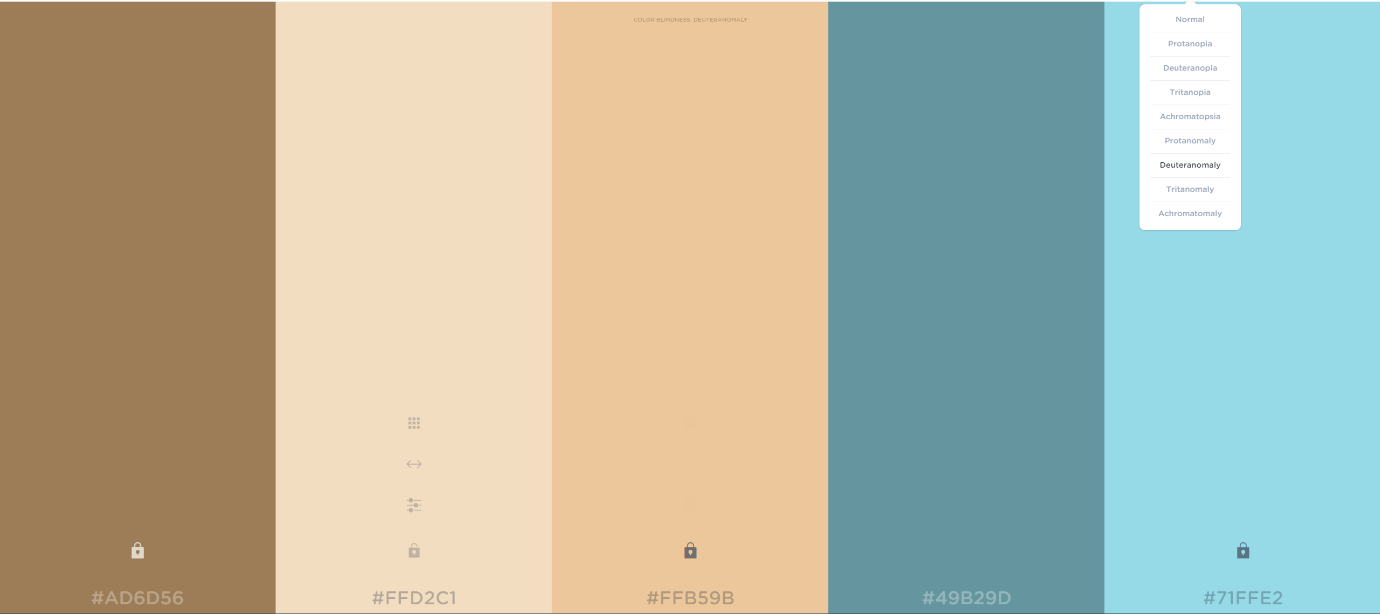
Achromatopsia



Protanomaly



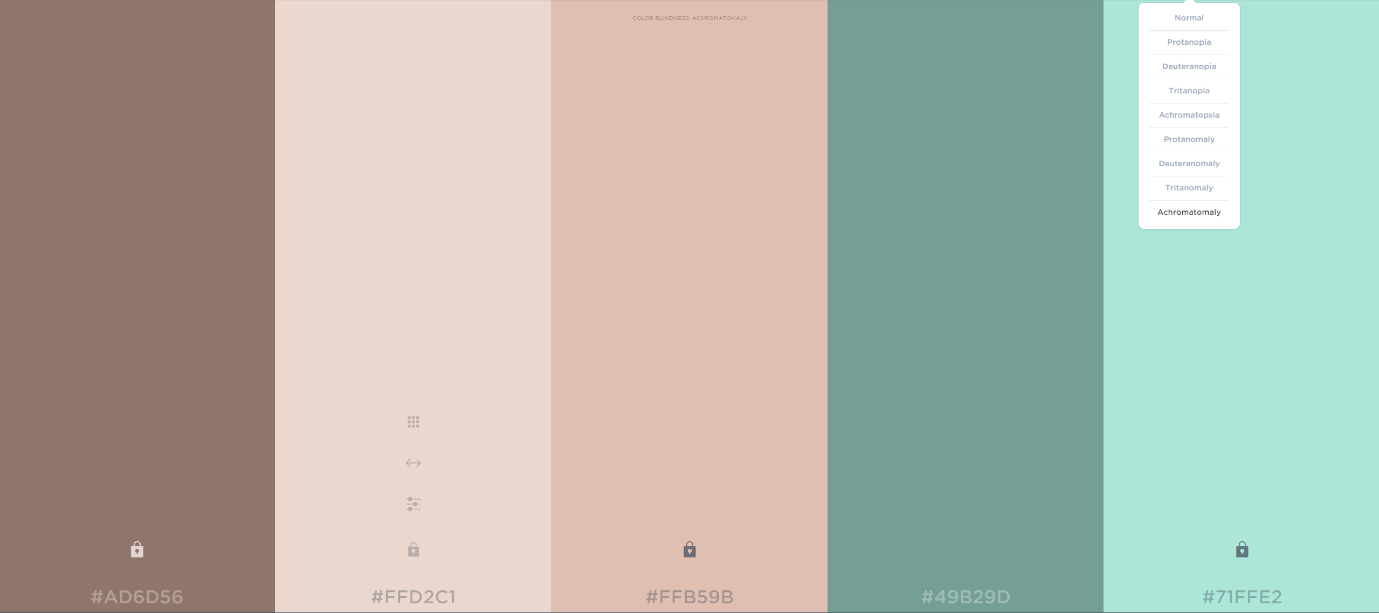
Deuteranomaly



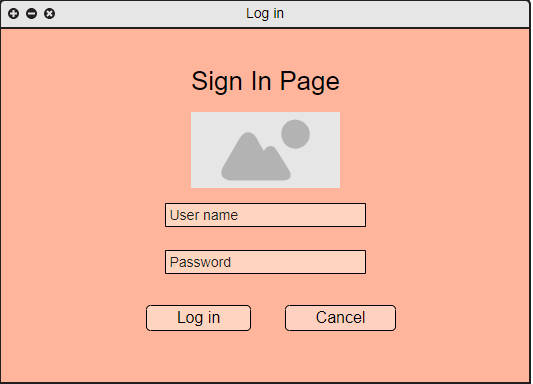
Tritanomaly

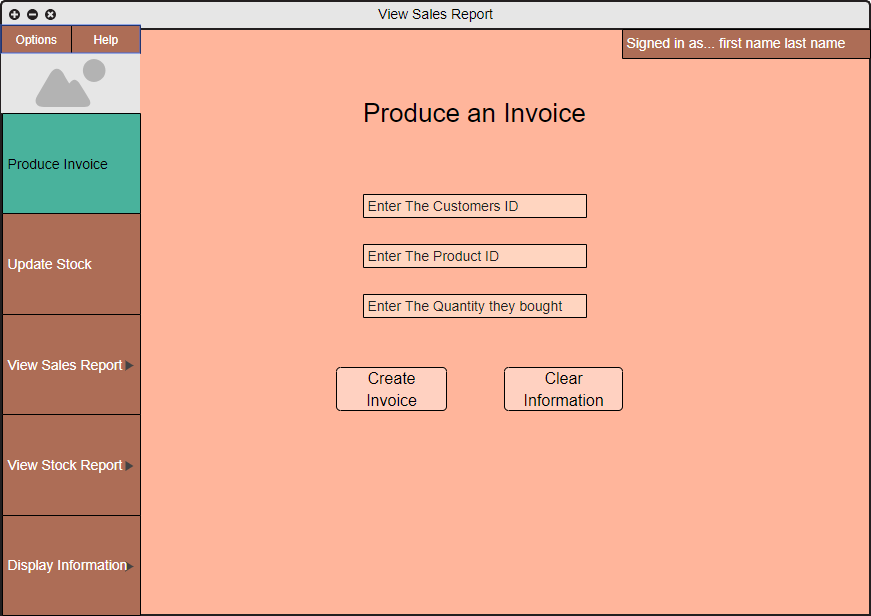


Achromatically



**1.8.3.3 User interface colour scheme 3**





#### 1.8.4 User interface client questionnaire

I gave the client a questionnaire that mainly focused on what they thought about the user interface and to get any feedback that about the interface I showed the client the wireframes that I created and the three different colour schemes that I created.

Question 1: on a scale from one to ten how you like the lay out of the wire frames for the application

10

Question 2: on a scale from one to ten how do you like the way that the application is navigated

10

Question 3: on a scale from one to then how to you like the way that the reports display the information

8

Question 4: on a scale from one to ten how do you like the colour scheme for colour scheme 1

10

Question 5: on a scale from one to ten how do you like the colour scheme for colour scheme 2

9

Question 6: on a scale from one to ten how do you like the colour scheme for colour scheme 3

5

Question 7: out of the three options which of the colour schemes did you like the most

Option 1

Question 8: if the answer to question 6 was none what did you not like about the colour schemes

N/A

Question 9: is there anything you would like changed about the colour scheme that you picked

nope

Question 10: do you have any additional feedback

great design follow through this with your development.

### 1.9 Client interview

On the 27 of February I had the second meeting with the client to clarify anything that I was this interview mainly focused on any extra things that I could add to the user interface

Question 1: how would you like the instructions on how to use the application be displayed

Help information can be displayed on a website

Question 2: does the company have a logo that they would like to be used in the application

Nope – you can create

Question 3: if you do not have a logo would you like one made up

yes

Question 4: what information would you like to be displayed in the transaction logs

Create a dummy transaction log

Question 5: During the inception phase meeting you asked about a feature that would give a warning for low stock how would you like this feature implemented suggestion (green if item above 3) yellow if 3 or less red if 0)

Yes, this is perfect

Question 6: would you like the system to be able to save the reports

Save reports in data base

#### 1.9.1 Colour scheme decision

After having a discussion with the client about the colour scheme that will be used in the application I have decided that moving forward the application will be using option 1 for the colour scheme I will be expanding upon the initial wire frames that I created to include a view of what each page in the application will look like with the chosen colour scheme.

# **2 Design**

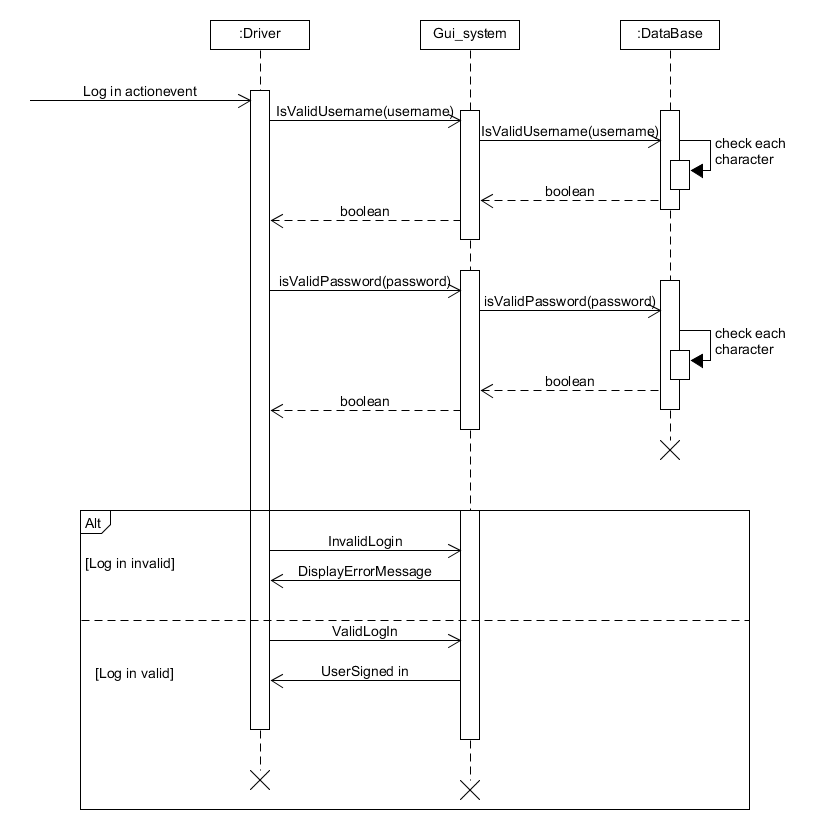
## Business model design

## 2.1 Dynamic model

### 2.1.1 Sequence diagrams

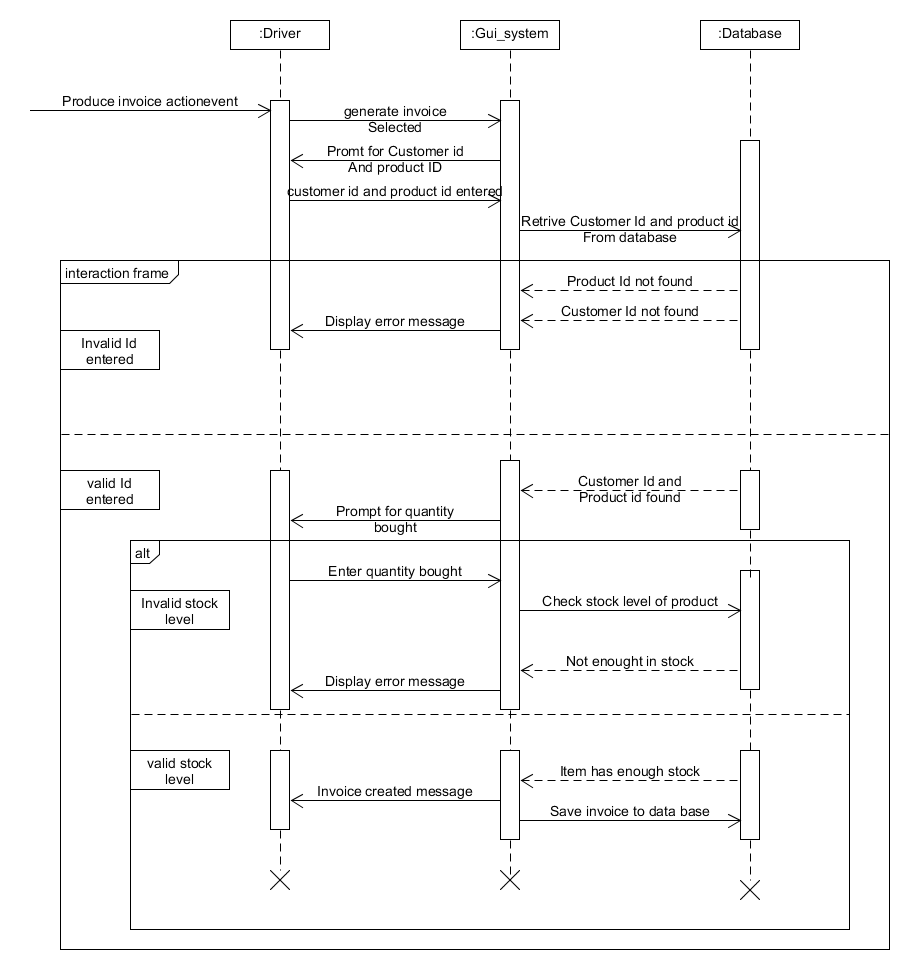
The reason that I am using sequence diagrams is to help me get an idea of what objects will need to communicate with each other. It also shows what order each of the interacts between the objects happens using these diagrams also can help to show what will happen in a few different scenarios for example if invalid data is given it will display an error message and if correct information is given the diagram will continue as normal the sequence diagrams that I am creating are based on the use case descriptions that I made during section 1.4.

#### Log in page diagram



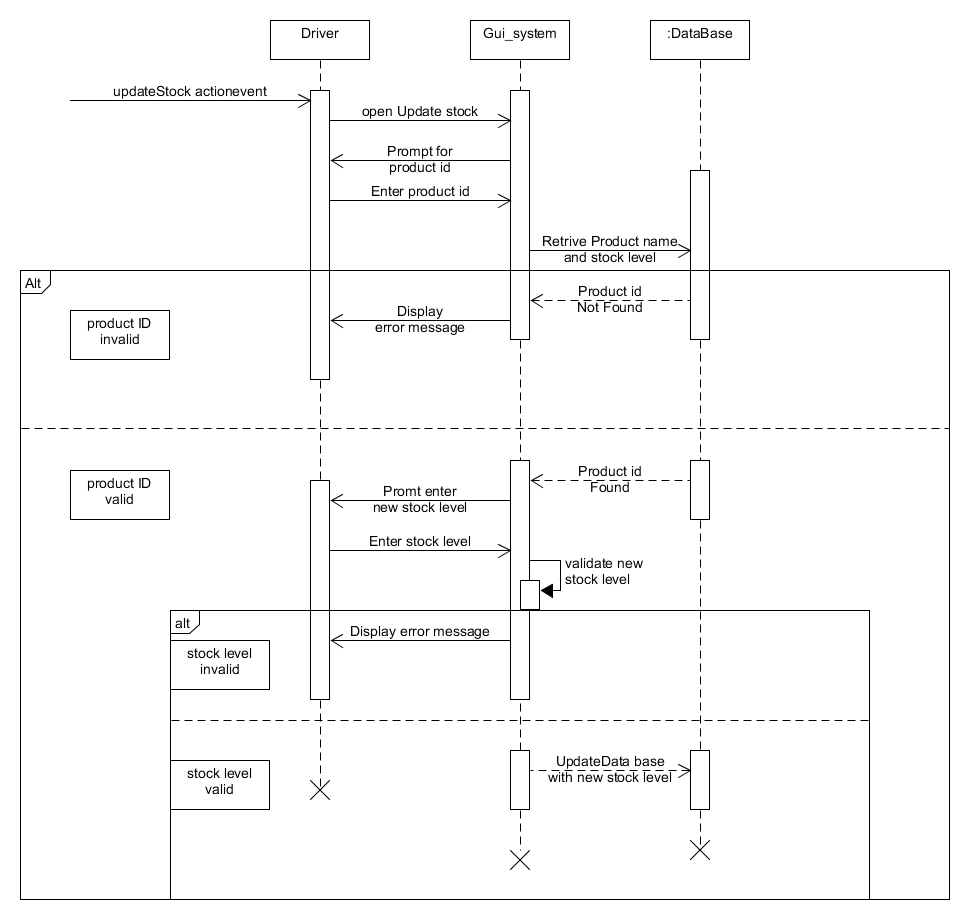
this diagram shows how objects will interact with each other during the log in function of the system it will also show what happens if invalid information is given.

#### Produce invoice sequence



This diagram will show the interactions between the objects for the creation of the invoices that the system will carry out it shows the errors that will happen if invalid information is entered by the user.

#### Sequence Diagram update stock



This diagram will show the interactions between the objects for the creation of the invoices that the system will carry out it shows the errors that will happen if invalid information is entered by the user.

#### Produce report

#### 

This diagram will show the interactions between the objects for the creation of the reports that the system will carry out it shows the errors that will happen if invalid information is entered by the user

#### Display information

#### 

This diagram shows how objects will interact with each other and how the information will be retrieved and then sent back to the system when the user presses the display information button

### 2.1.2 Activity diagrams

I am also using activity diagrams to show the flow of how each of the functions will be carried out and how object interact with each other when the function is being carried out. The diagrams also show decisions that the application has to make such as if the user has given invalid information what It should do then I will be creating these diagrams based on the use cases that I have already identified in section 1.4 during the analysis stage.

#### Activity diagram log in

#### 

Activity diagram what shows the flow for the user log in it shows what happens if the user gives invalid information and what happens if valid data is entered the diagram also shows how the classes interact with each other during the log in action.

#### Activity diagram produce invoice

#### 

This activity diagram shows the flow for the creation of an invoice and what happens if the user enters invalid information it also shows how the classes in the system interact with each other when the action is being carried out.

#### Activity diagram generate report

#### 

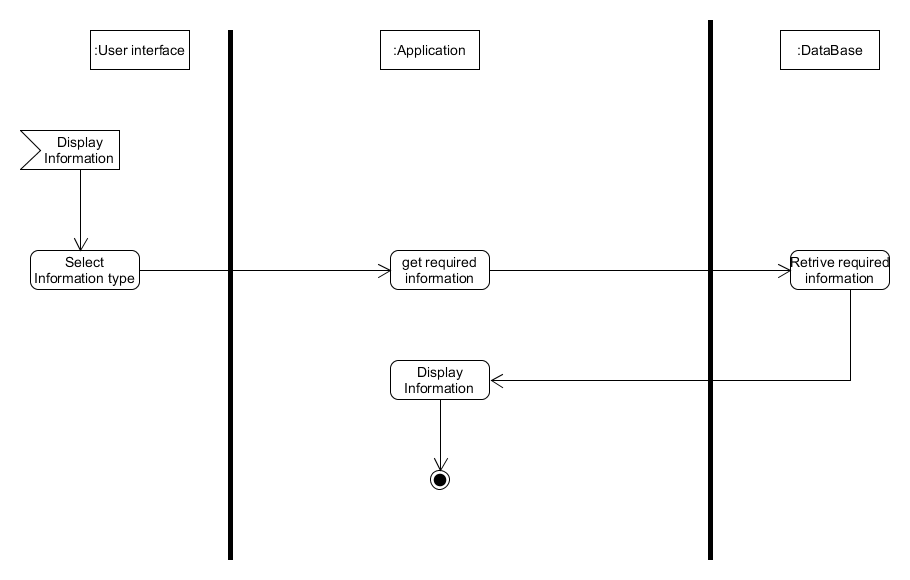
This activity diagram shows the actions that happen during the generation of a report it shows how the classes interact with each other and what happens if there is invalid data entry.

#### Activity diagram update stock

#### 

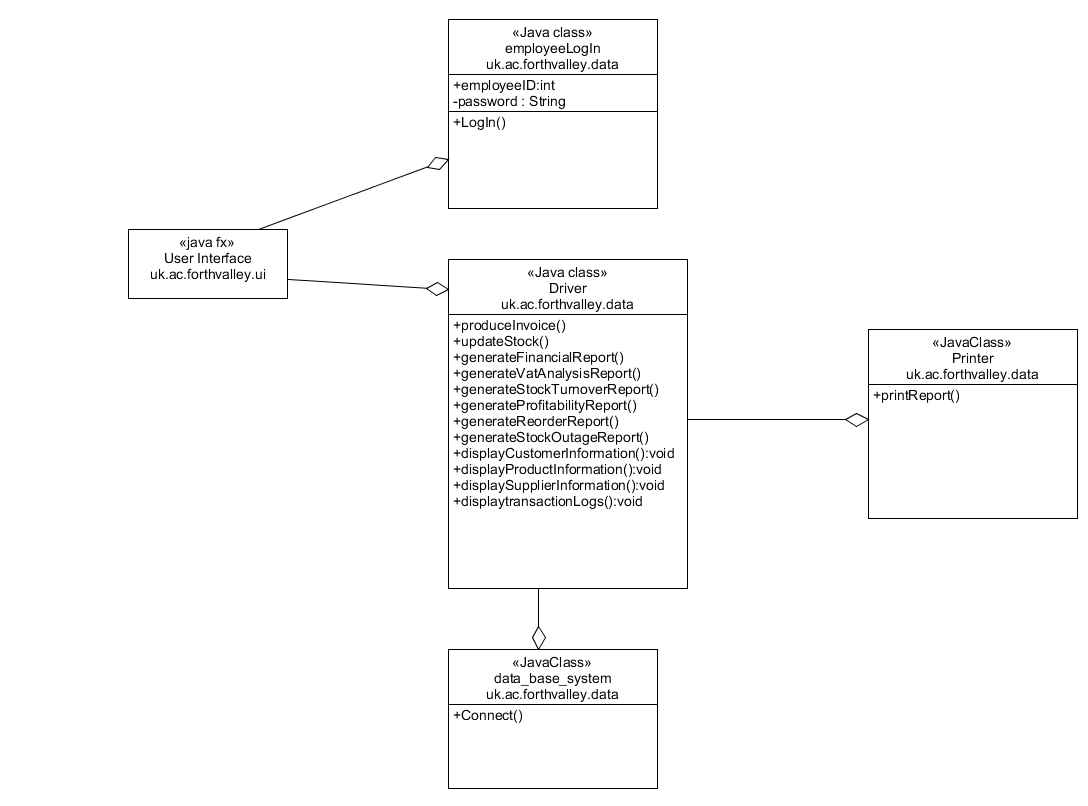
This activity diagram shows what actions the update stock function will have to carry out in order to up date the stock levels it shows how the classes interact with each other and what will happen if invalid data is entered

#### Activity diagram display information



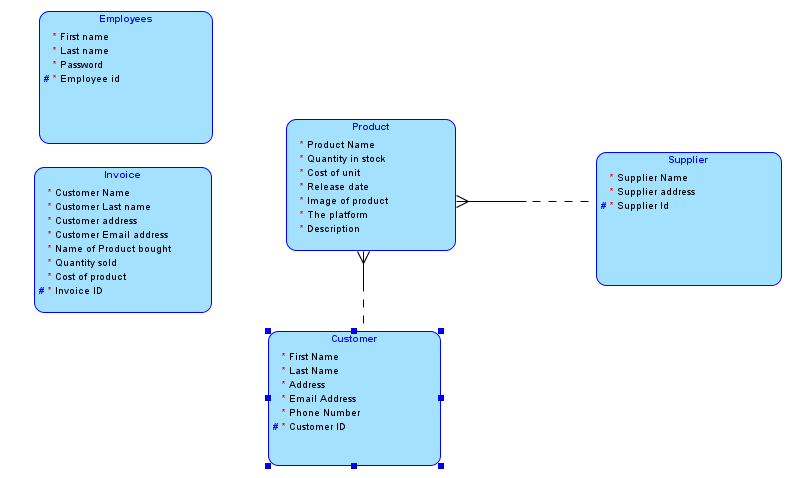
This activity diagram shows how the information display action will work in the system it shows how the classes interact with each other and each step the system must take for the information to be displayed.

## 2.3 Class diagram - finalized



I have expanded upon the initial class diagram that I create this diagram gives more detail on what methods each of the classes will have in the program and their visibility settings and how each part of the system will communicated with each other.

## 2.4 ERD - Finalized



Expanding upon the initial erd that I have created in section 1.6.1 I have now added the attributes to the erd and if they are mandatory or not the diagram also now shows the primary keys that the data base will use, I have also created a table that shows what data type each of the attributes will use.

### 2.4.1 Data table

I have created these tables for when I am building the data base of the system, I will have an idea of what attributes will be used in each table and their data types these tables will also be used when I am showing the data binding model.

**Employees**

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Data Type** | **Optionality** |
| First name | Varchar2 (20) | Mandatory |
| Last name | Varchar2 (30) | Mandatory |
| Password | Varchar2 (25) | Mandatory |
| Employee id | Integer | Mandatory |

**Products**

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Data Type** | **Optionality** |
| Product Name | Varchar2 (20) | Mandatory |
| Quantity in stock | Number (3) | Mandatory |
| Cost of unit | Number (5,2) | Mandatory |
| Release date | date | Mandatory |
| Image of product | raw | Mandatory |
| The platform | Varchar2 (10) | Mandatory |
| Description | Varchar2 (100) | Mandatory |
| Product id | Integer | Mandatory |

**Customer**

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Data Type** | **Optionality** |
| First name | Varchar2 (20) | Mandatory |
| Last name | Varchar2 (30) | Mandatory |
| Address | Varchar2 (40) | Mandatory |
| Email address | Varchar2(40) | Mandatory |
| Phone Number | Varchar2 (12) | Optional |
| Customer id | Integer | Mandatory |

**Supplier**

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Data Type** | **Optionality** |
| Supplier name | Varchar2 (25) | Mandatory |
| Supplier address | Varchar2 (30) | Mandatory |
| Supplier id | Integer | Mandatory |

**Invoice**

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Data Type** | **Optionality** |
| Customer name | Varchar2 (25) | Mandatory |
| Customer Last name | Varchar2 (30) | Mandatory |
| Customer address | Varchar2 (25) | Mandatory |
| Customer email address | Integer | Mandatory |
| Name of product bought | Varchar2 (20) | Mandatory |
| Quantity sold | Number (2) | Mandatory |
| Cost of product | Number (5,2) | Mandatory |
| Invoice id | Integer | Mandatory |

## View model design

## 2.5 Finalized user interface design

During the analysis stage in section 1.8 I had a discussion with client about deciding what colour scheme was going to be used and after that discussion I have decided that I am going to use colour scheme one for the user interface I have now added colour to each of the wire frames to show the client what each page will look like with colour. I have also added the images that I will be using as well.

I feel this choice of user interface was a good design choice as the interface is not cluttered with many bright colours the text is easy to read and It will be usable for people who suffer from colour blindness I feel the way that the navigation is handled will be easy for a new user to pick up and the information the application will display is easy to find.

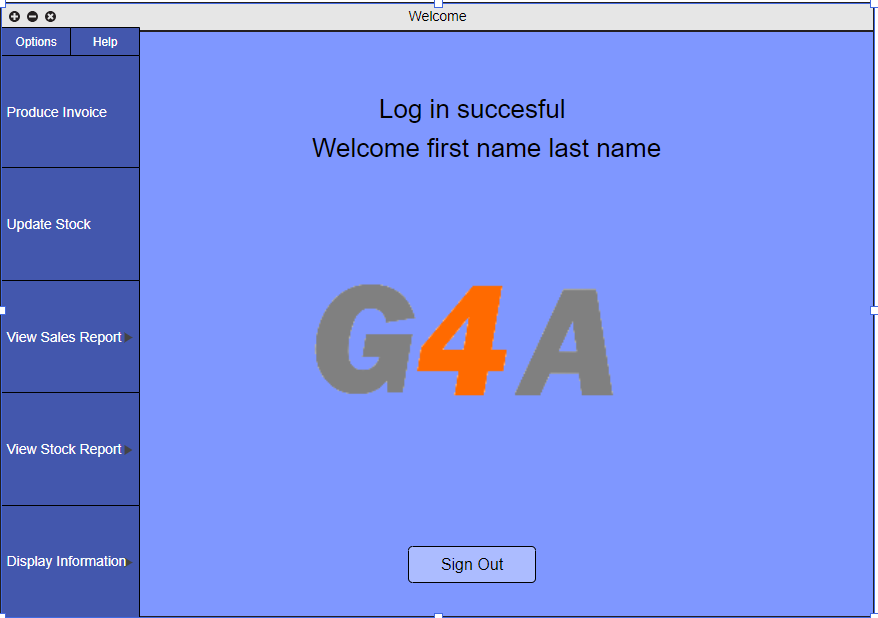
During the interview with the client I found out that the company does not have a logo so I have also designed a logo that will be used in the application the way that I designed the logo was to use paint.net.

**Log in page**



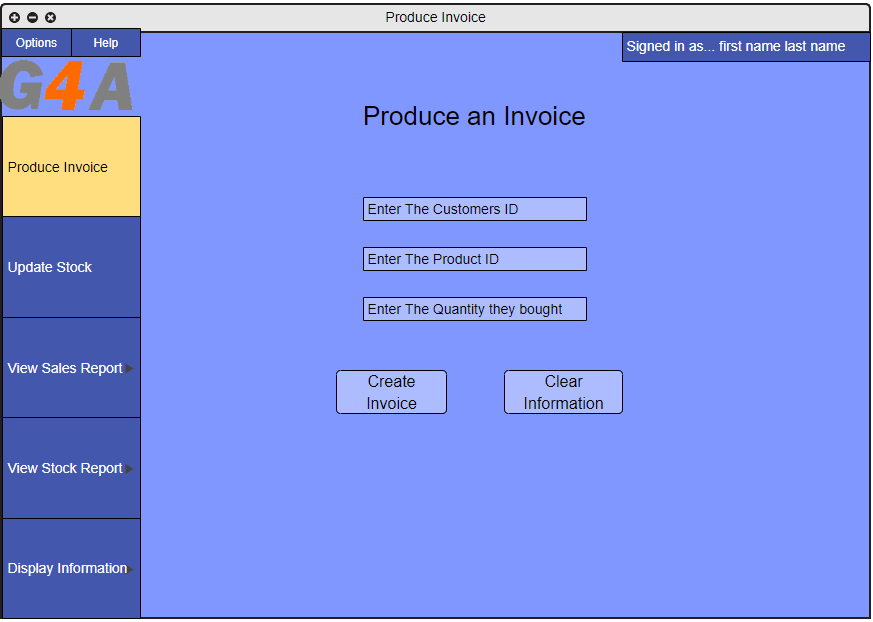
This wireframe shows the finalized design that I will be using for the login page of the system it will be the screen that is first displayed to the user.

**Welcome screen**



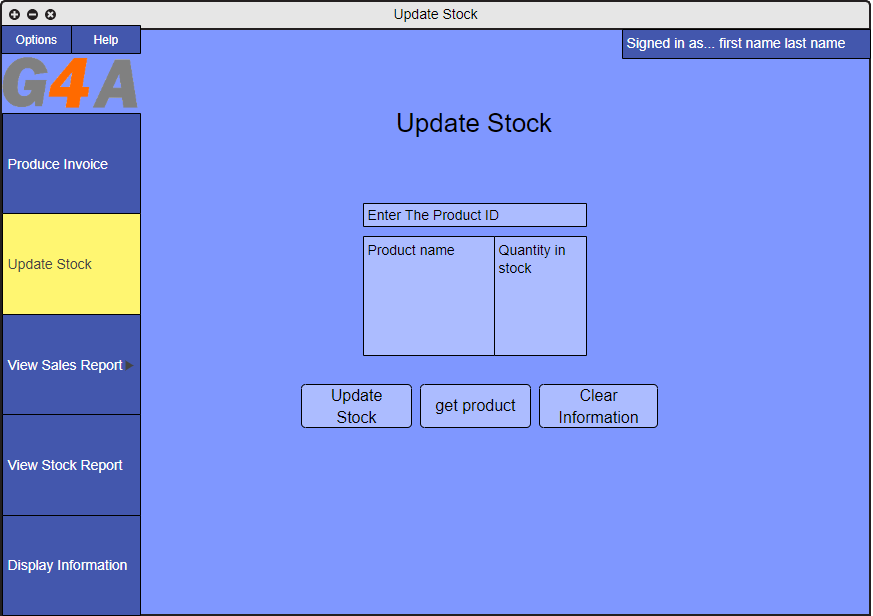
This wireframe shows the finalized design for the welcome screen interface this will be the screen displayed to a user after they successfully logged in the functions this page will be able to carry out is for the user to sign out which will return the user to the log in screen and also use it to navigate to the section of the application you want to use.

**Produce invoice**



This is the finalized design for the produce invoice screen in the application the function that this page will carry out is to allow the user to create an invoice for a purchase that has been made the way the page will function is for the employee to enter the customer and product id and quantity of the product the user purchased. The employee will then select the create invoice button to create the invoice also if they change their mind, they will be able to select the clear information button that will remove the text that is the text fields.

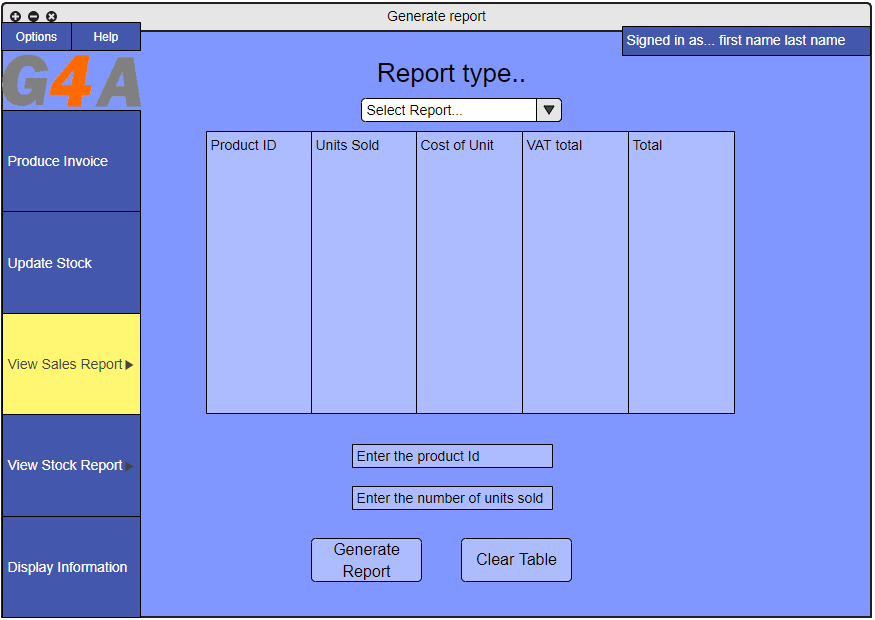
**Update stock**



This is the finalized design for the update stock interface this page will be used to display the stock of an item that the employee wants to view after entering the product id of the item once the user presses the get product button the name of the product and is stock level will be displayed to screen.

The user will then be able to press the update stock button that will bring up a menu for the user to enter the new stock level of the item selected.

**Generate report - financial**



This interface shows the finalized design for the generate of financial reports the way that this page will function is for the user to enter the id of the product they want the report about, and the number of units sold the system will then perform a calculation to get the total value of the number of units sold.

**Vat analysis**



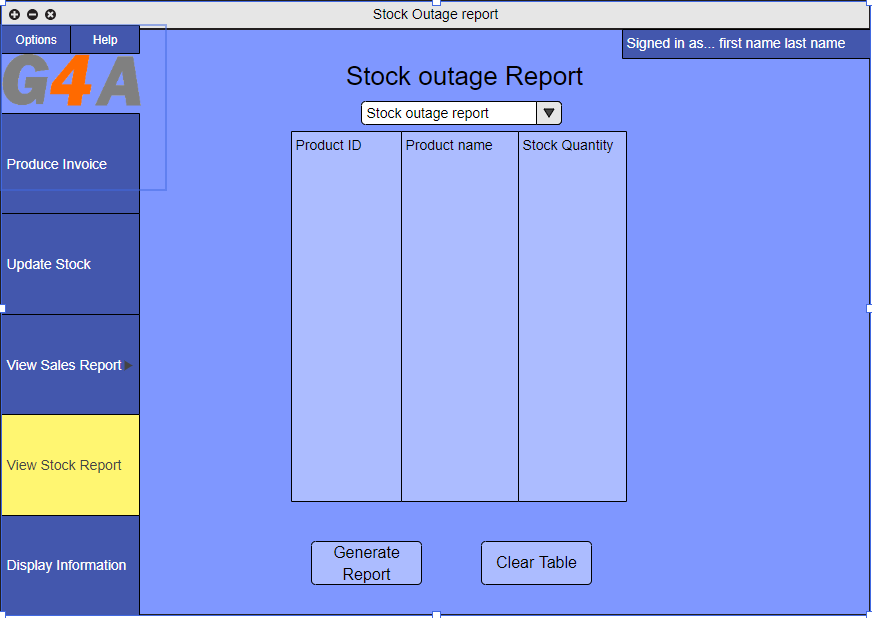
This page shows the finalized design for the vat analysis interface the way that this page will function is that the user will enter the id of the product, number of units sold and the cost of the product. The application will then calculate the vat of the products and display the total vat to screen.

**Probability report**

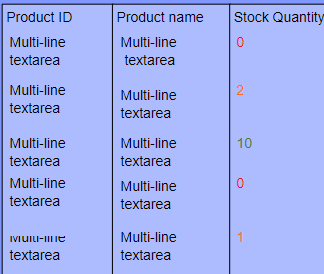


This page shows the finalized design for the probability report interface the employee will be able to generate a profitability report about a product they sell they will enter the product id and the purchase price of the item.

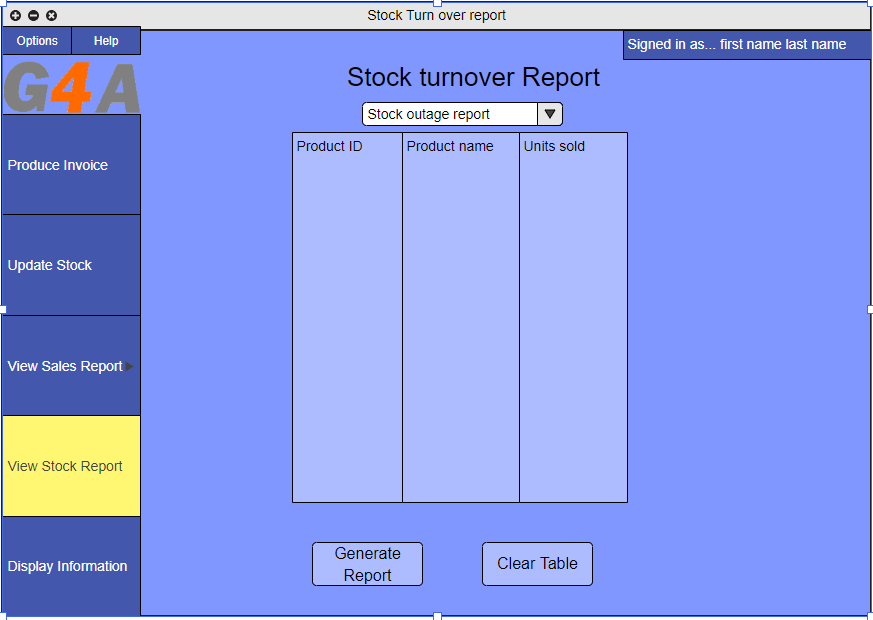
**Generate report – stock outage**



This page shows the finalized design for the generation of a stock report interface the user will select the generate report button and a list of all the products will be displayed with their stock levels each stock level will be categorized with a different colour red will mean that the stock level is at 0 , yellow will mean that the stock level is between 1 and 3 and green means that the stock level is above 3 an example of this is displayed in the page below.

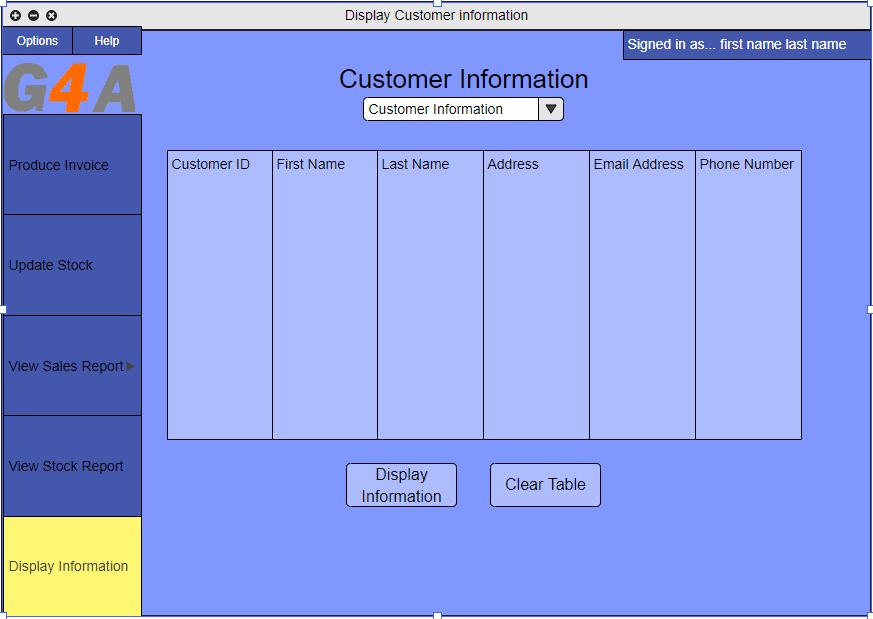


**Stock turnover report**



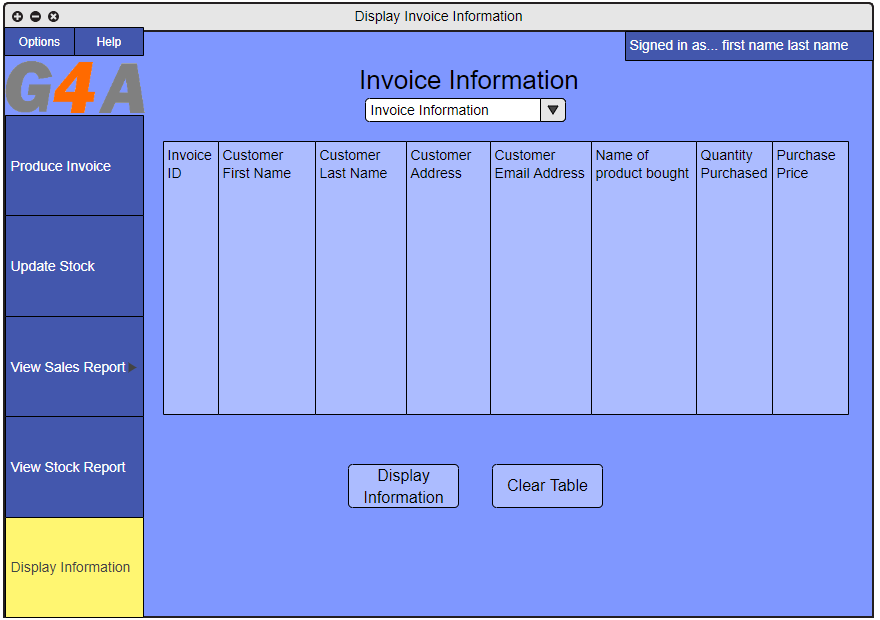
This page shows the finalized design for the stock turnover report interface this page will be used to display information about how many units of a product the company has manged to sell

**Display information - Customer**



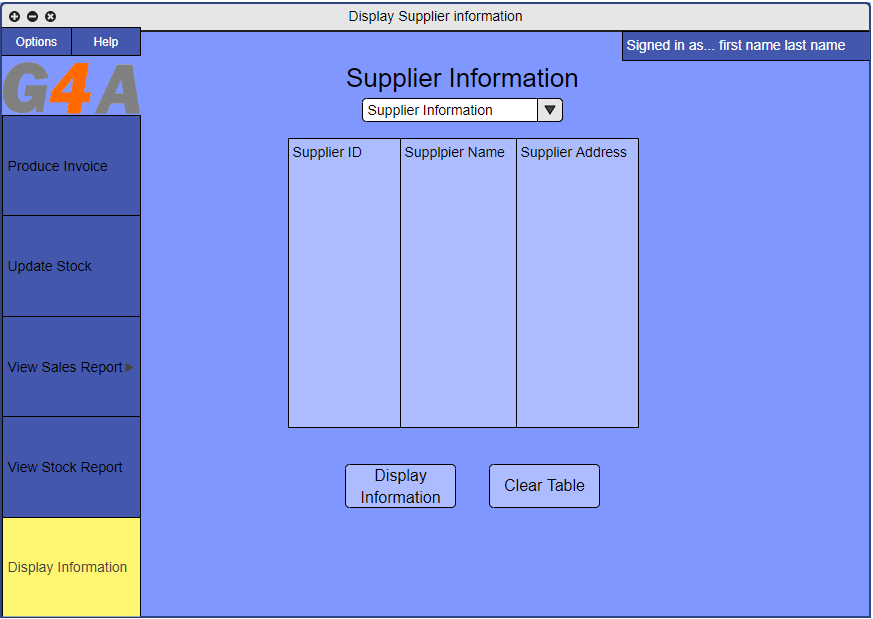
This is the finalized design for the display customer information page the employee will be able to press the display information button and information about the customers they have in their system will be displayed.

**Display information - Invoice**



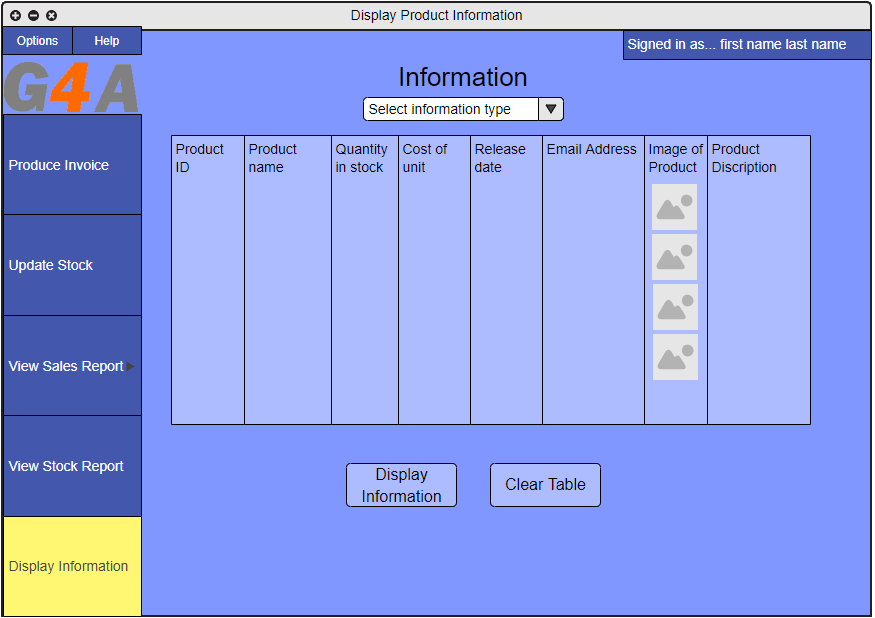
This page is the finalized design for the display information about invoices page the employee will be able to press the display information button and a list of all the invoices that they have in their system will be displayed.

**Display information - supplier**



This page is the finalized design for the interface being used to display information about suppliers the employee will be able to display information about any of their suppliers that they have stored.

**Display information - product**

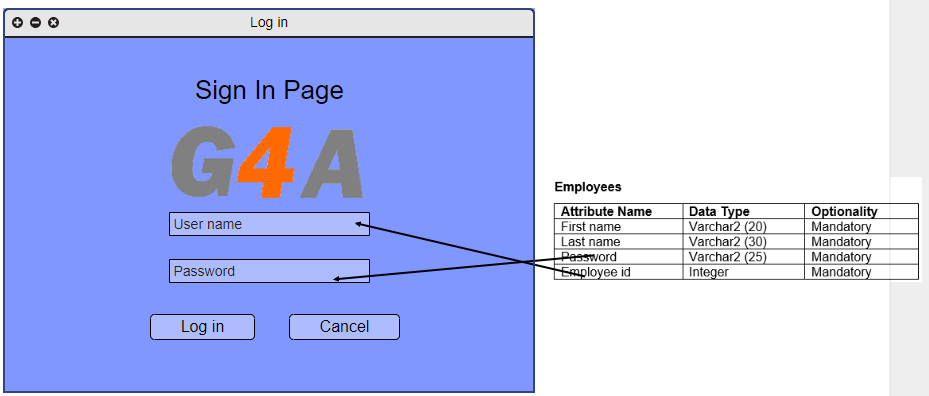


This page is the finalized design for the interface that will be used to display the information about products the way that this page will function is that the employee will select the display information button and a list about all the products that they have in their system will be displayed to screen.

## 2.6 Data binding model

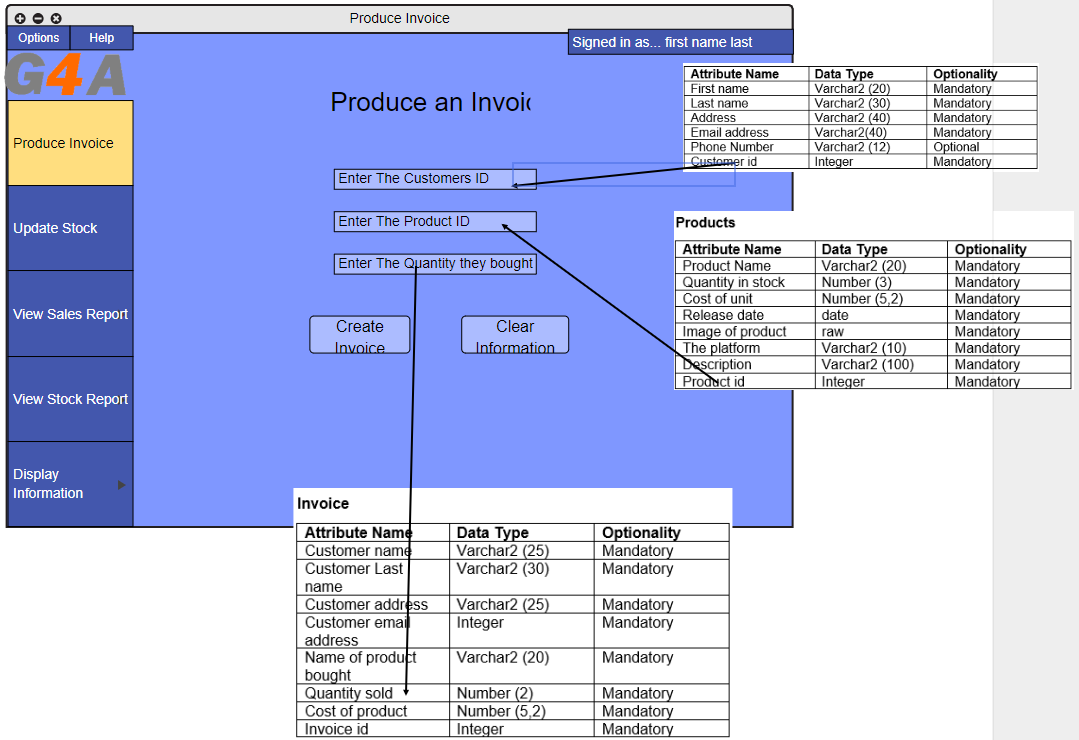
Using the data table that I created for each of the entities (section 2.4.1) I have created diagrams that use the technique that I had identified (section 1.7) I will be able to show how the data binding will work in the system these diagrams will show when a user enters a value what the system will do to collect or store the information from the data base

**Log in page**



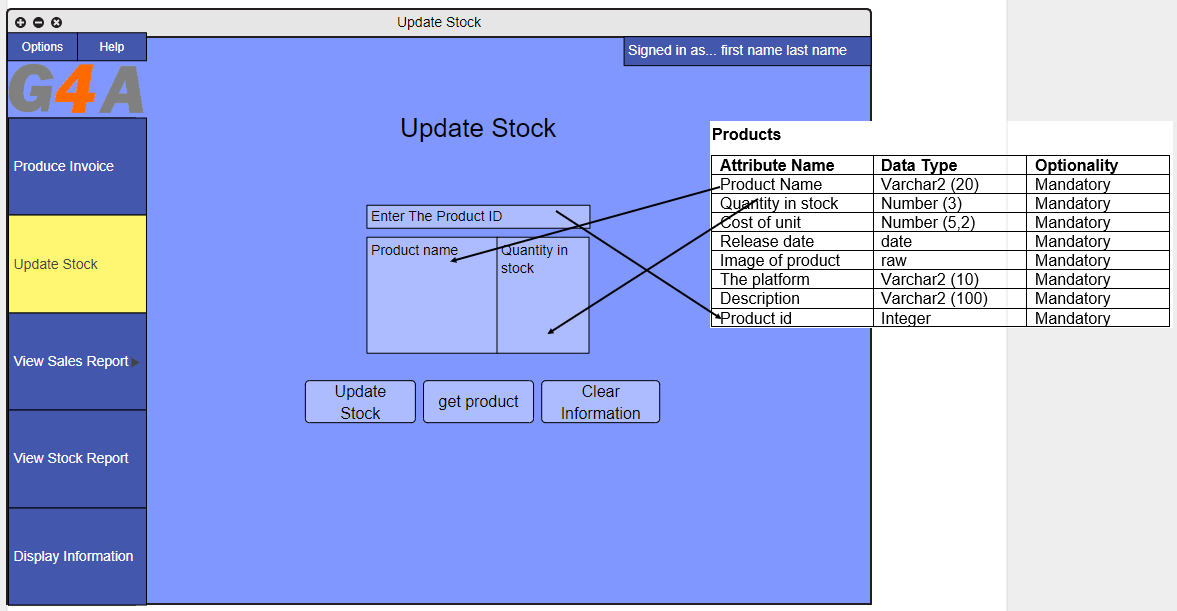
This diagram shows what information the user will need to enter in order to log in and where the information will be retrieved from employees will have their log in data stored in the data base and whenever an employee logs in the program will use their employee id to search if they are in the system and if they are retrieve their information. If the data base is able finds their information in the system, they will be successfully logged in. if the system doesn’t find their information, they will be given an error message telling the user that they have entered invalid information.

**Produce invoice**



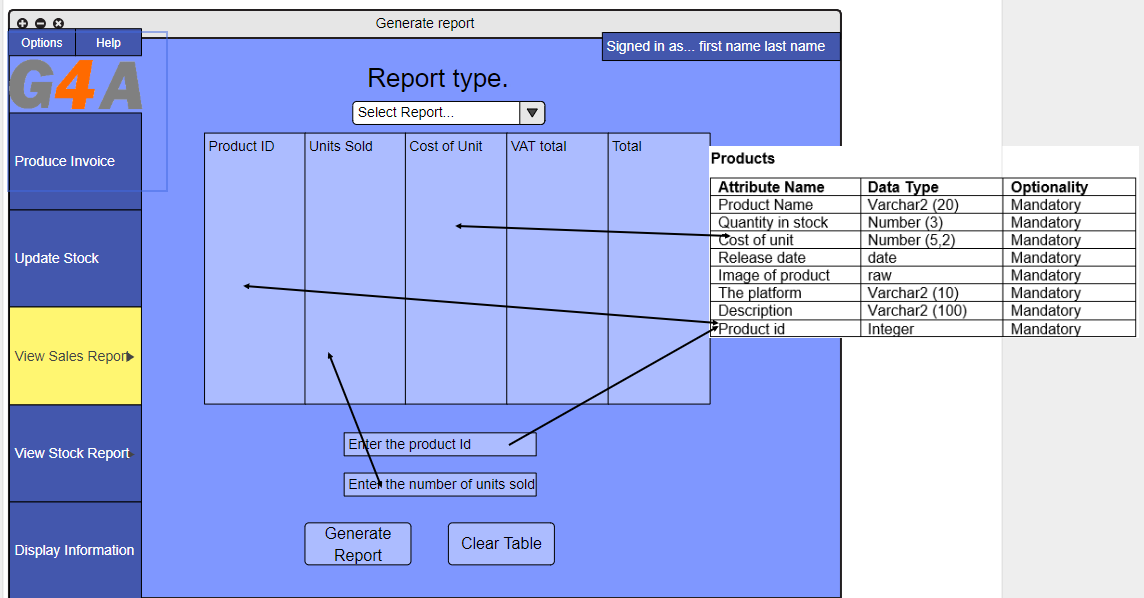
This diagram shows how the application will retrieve and store information about the invoices produced the application will retrieve the customers product id and customer id and then store the quantity bought in the invoice table. The way the rest of the invoice will be generated is by taking the information from the customer and product table that is required for the invoice and storing that information in the invoice table. If any of the information entered by the user is invalid an error message will be displayed

**Update Stock**



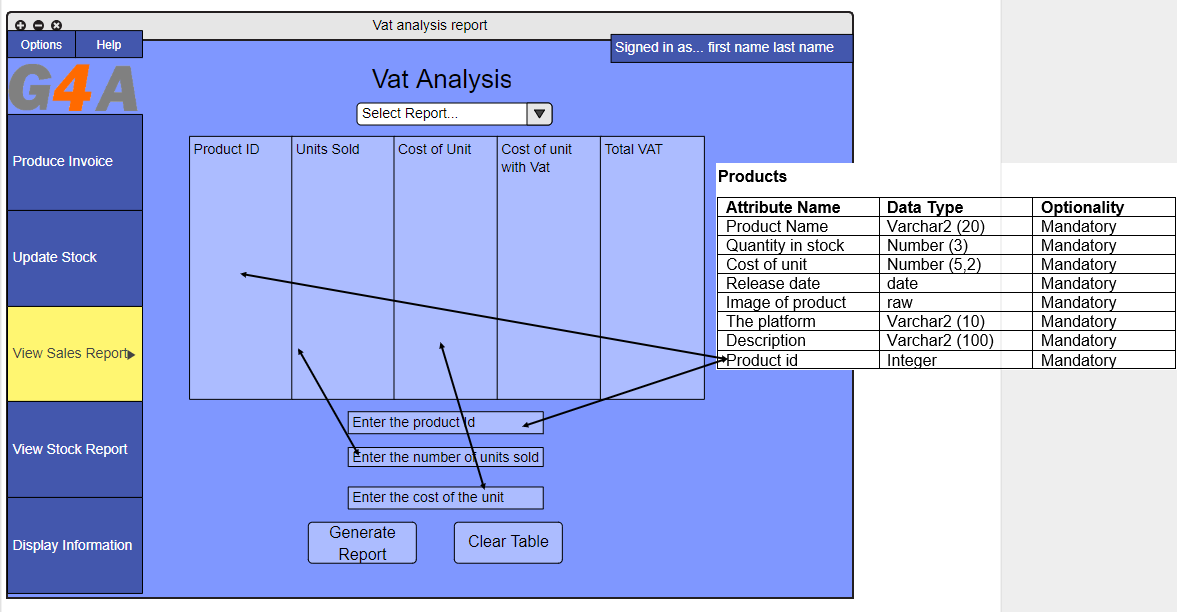
The way this page will bind the data is that whenever the user enters the id of a product it will be retrieved from the data base and the information required will be displayed to the screen the user will then be able to enter the new stock quantity to update the data base if no product id is found an error message will be displayed.

**Generate report - financial**



The way this page will handle data binding is that the user will enter the information that is required and the system will then look up the id of the item to retrieve the required information about the item and also the user will enter the number of units that have been sold the system will then retrieve the information that is required from the data base the reports will be then saved in a txt file for later viewing. If the user enters invalid information the system will display an error message.

**Vat analysis**



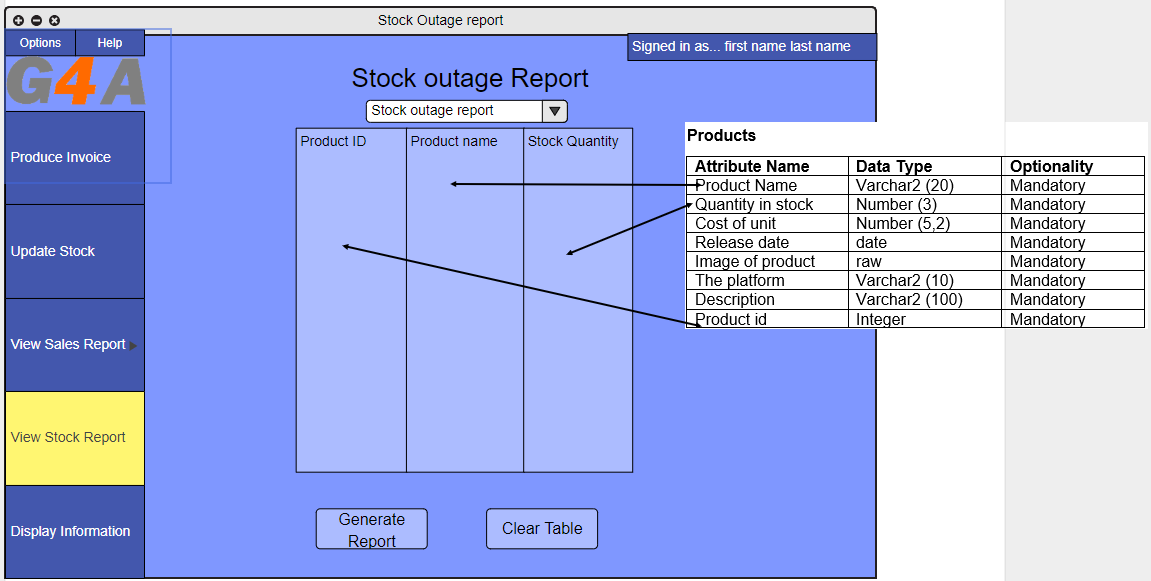
The vat analysis will retrieve the information about the product required from the data base and will it will function in a similar way to the generate of the financial reports expect that rather than getting the total sales of a game it will calculate how much vat is on a sale.

**Profitability report**



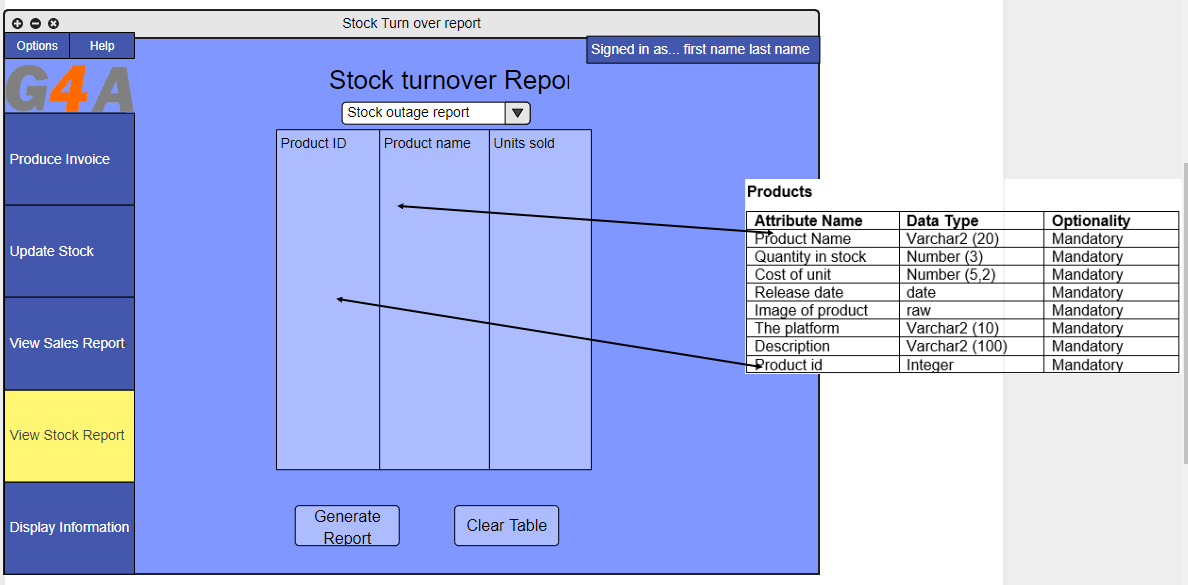
The profitability reports will retrieve information about the product from when the user enters the product ID the system will then calculate the price that an item was purchased for and then display the difference.

**Stock outage report**



The stock outage report will retrieve information about all the products in the system that have a stock level of zero it will user the products table to help identify this

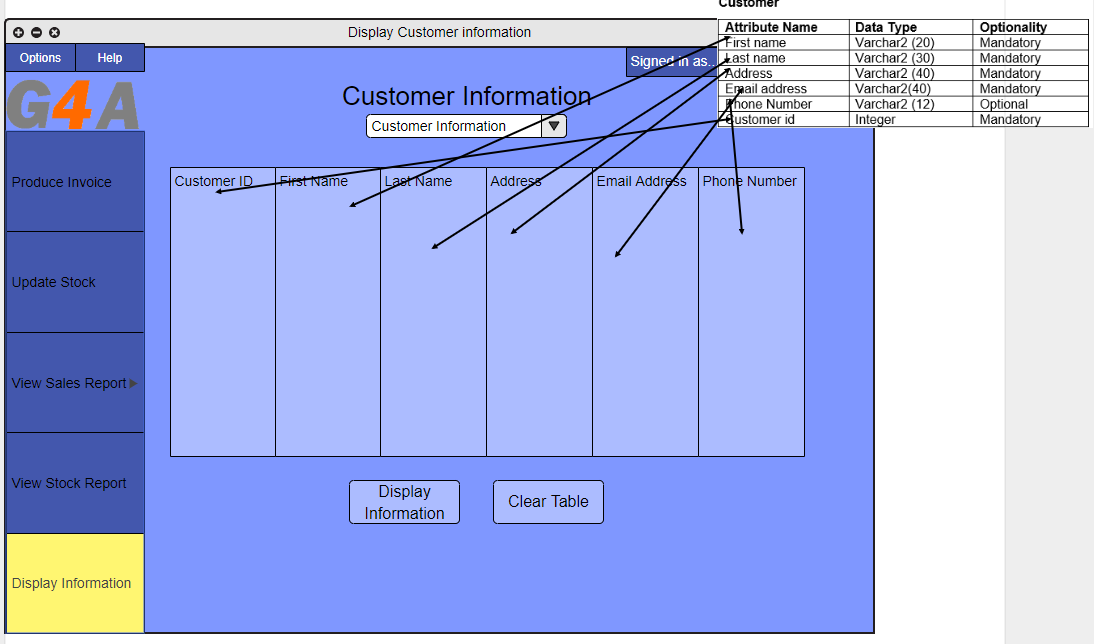
**Stock turnover report**



**Display information**

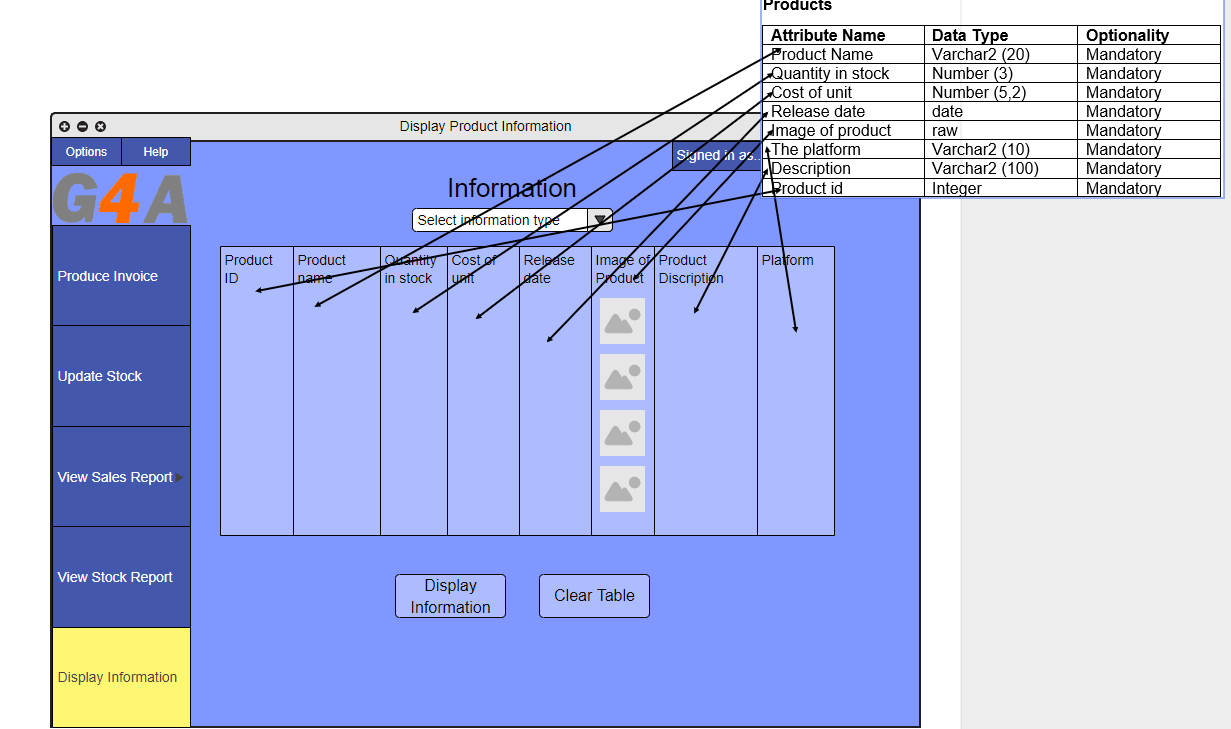
The models used for the display information are all the same apart from what table of the data base the information is collected from once the user clicks on the display information button.

**Customer information**



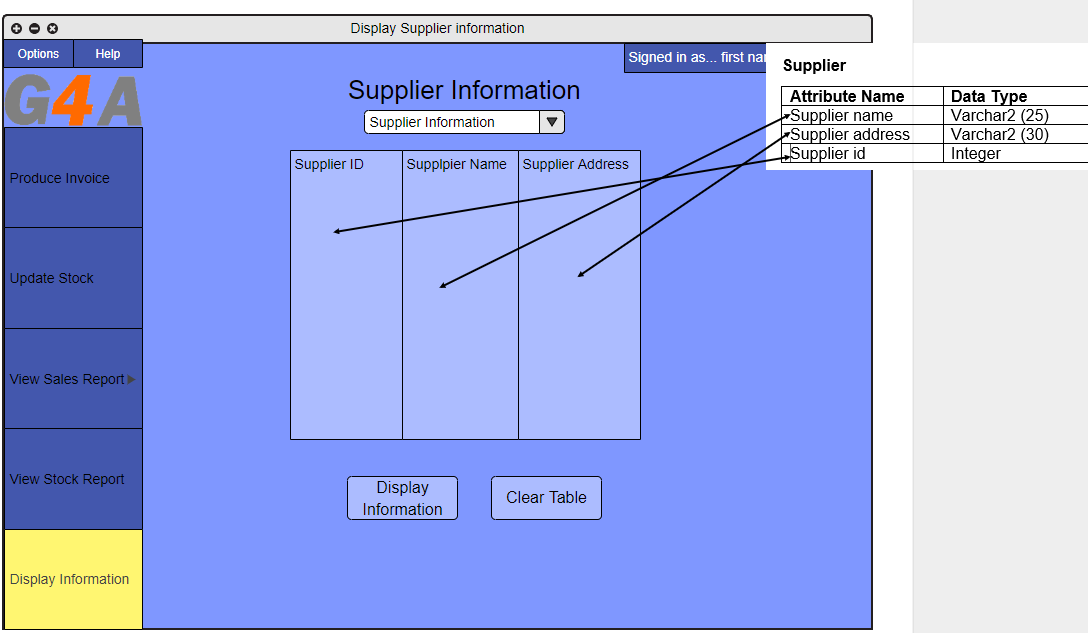
This diagram shows how the customer information that is being retrieved. It will be retrieved by the system when the user presses the display information button it will take a list of the full customer table and then display it to screen.

**Product information**



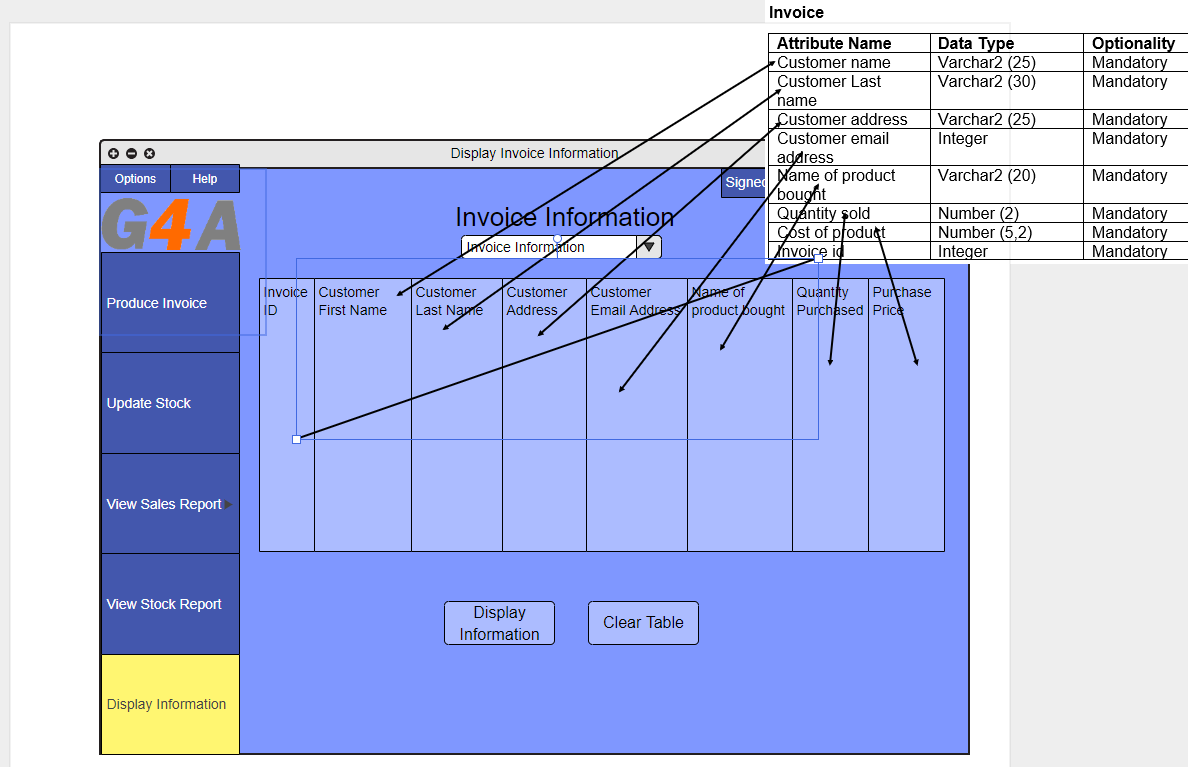
This diagram shows how the product information that is being retrieved will be retrieved by the system when the user presses the display information button it will take a list of the full customer table and then display it to screen.

**Supplier information**



This diagram shows how the supplier information that is being retrieved will be retrieved by the system when the user presses the display information button it will take a list of the full customer table and then display it to screen.

**Invoice information**



This diagram shows how the invoice information that is being retrieved will be retrieved by the system when the user presses the display information button it will take a list of the full customer table and then display it to screen.

# Bibliography

Adobe. (2019, 03 02). *Adobe color cc*. Retrieved 03 02, 2019, from color.adobe.com: 0.6784313725490196,0.42745098039215684,0.33725490196078434,1,0.8222733517597964,0.7584028375485778,1,0.711754785859582,0.608166662028039,0.2857166634196273,0.7,0.6155183901423315,0.44239518618293205,1,0.8862919452071362

Collinge, R. (2017, 01 17). *How to Design for Color Blindness*. Retrieved 02 26, 2019, from usabilla.com: https://usabilla.com/blog/how-to-design-for-color-blindness/

Interaction design foundation. (2019, 03 02). *User Interface (UI) Design*. Retrieved 02 26, 2019, from www.interaction-design.org: https://www.interaction-design.org/literature/topics/ui-design

Usability.gov. (2019, 03 02). *User Interface Design Basics*. Retrieved 03 02, 2019, from www.usability.gov: https://www.usability.gov/what-and-why/user-interface-design.html