

## Evidence Gathering Document for SQA Level 8 Professional Developer Award.

This document is designed for you to present your screenshots and diagrams relevant to the PDA and to also give a short description of what you are showing to clarify understanding for the assessor.

Each point that required details the Assessment Criteria (What you have to show) along with a brief description of the kind of things you should be showing.

Please fill in each point with screenshot or diagram and description of what you are showing.

### Week 2

Unit	Ref	Evidence
I&T	I.T.5	Demonstrate the use of an array in a program. Take screenshots of: *An array in a program *A function that uses the array *The result of the function running
		<b>Description:</b> <ul style="list-style-type: none"><li>- An array of train station stops</li><li>- Unshift function used to add Glasgow Queen St to the beginning of the array</li><li>- The result of the function with Glasgow Queen St in the array</li></ul>

### Paste Screenshot here

```
stops = [ "Croy", "Cumbernauld", "Falkirk High", "Linlithgow", "Livingston", "Haymarket" ]
```

```
stops.unshift("Glasgow Queen St")
```

```
[→ hash_array_homework git:(master) ✘ ruby array_exercise.rb
["Glasgow Queen St", "Croy", "Cumbernauld", "Falkirk High", "Linlithgow", "Livingston", "Haymarket"]]
```

Unit	Ref	Evidence
I&T	I.T.6	Demonstrate the use of a hash in a program. Take screenshots of: *A hash in a program *A function that uses the hash *The result of the function running
		<b>Description:</b> <ul style="list-style-type: none"> <li>- Hash of a dog in a pet shop array</li> <li>- Test for the function to find the pet hash by key name</li> <li>- Test that if the name entered is not represented by the name key the function will return nil</li> <li>- Function to find the pet hash by the key name</li> <li>- Result of tests passing</li> </ul>

Paste Screenshot here

```
{
  name: "Arthur",
  pet_type: :dog,
  breed: "Husky",
  price: 900,
},
```

```
def test_find_pet_by_name_returns_pet #(full pet)
  pet = find_pet_by_name(@pet_shop, "Arthur")
  assert_equal("Arthur", pet[:name])
end

def test_find_pet_by_name_returns_nil
  pet = find_pet_by_name(@pet_shop, "Fred")
  assert_nil(pet)
end
```

```
def find_pet_by_name(pet_shop, name)
  for pet in pet_shop[:pets]
    if pet[:name] == name
      return pet
    end
  end
  return nil
end
```

```
[→ pet-shop_start_point git:(master) ✘ ruby specs/pet_shop_spec.rb
Run options: --seed 35210

# Running:

..
Finished in 0.001034s, 1934.2361 runs/s, 1934.2361 assertions/s.

2 runs, 2 assertions, 0 failures, 0 errors, 0 skips
```

## Week 3

Unit	Ref	Evidence	
I&T	I.T.3	Demonstrate searching data in a program. Take screenshots of: *Function that searches data *The result of the function running	<b>Description:</b>

Paste Screenshot here

```
def find_song(title)
  @songs.each do |song|
    if song.title == title
      return song
    end
  end
  return nil
end
```

```
def test_find_song_by_title()
  result = @room.find_song(@guest1.fav_song())
  assert_equal(@song2, result)
end

def test_find_song_by_title_false()
  result = @room.find_song("Not a Song")
  assert_nil(result)
end
```

```
➔ weekend_homework git:(master) ✘ ruby specs/room_spec.rb
Run options: --seed 54361

# Running:

..
Finished in 0.001094s, 1828.1536 runs/s, 1828.1536 assertions/s.

2 runs, 2 assertions, 0 failures, 0 errors, 0 skips
```

Unit	Ref	Evidence	
I&T	I.T.4	Demonstrate sorting data in a program. Take screenshots of: *Function that sorts data *The result of the function running	
		<b>Description:</b>	

**Paste Screenshot here**

```
numbers = [3, 5, 1, 4]

def sort_numbers(numbers)
  return numbers.sort
end

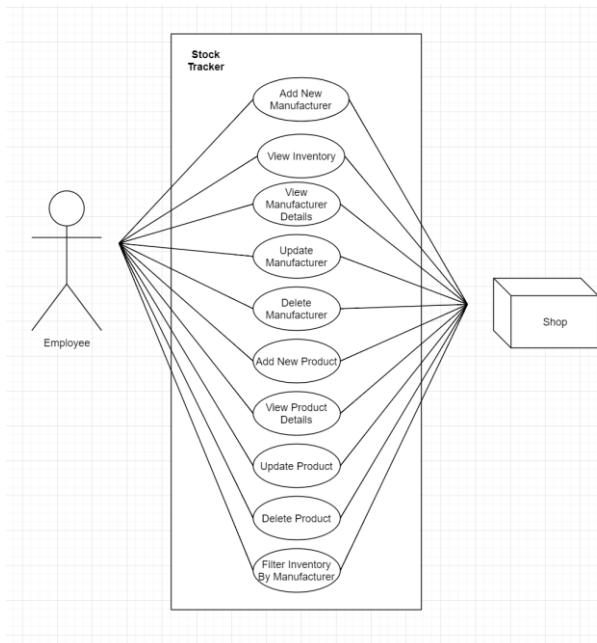
p sort_numbers(numbers)
```

→ **evidence** ruby extra.rb  
[1, 3, 4, 5]\_

**Week 5 and 6**

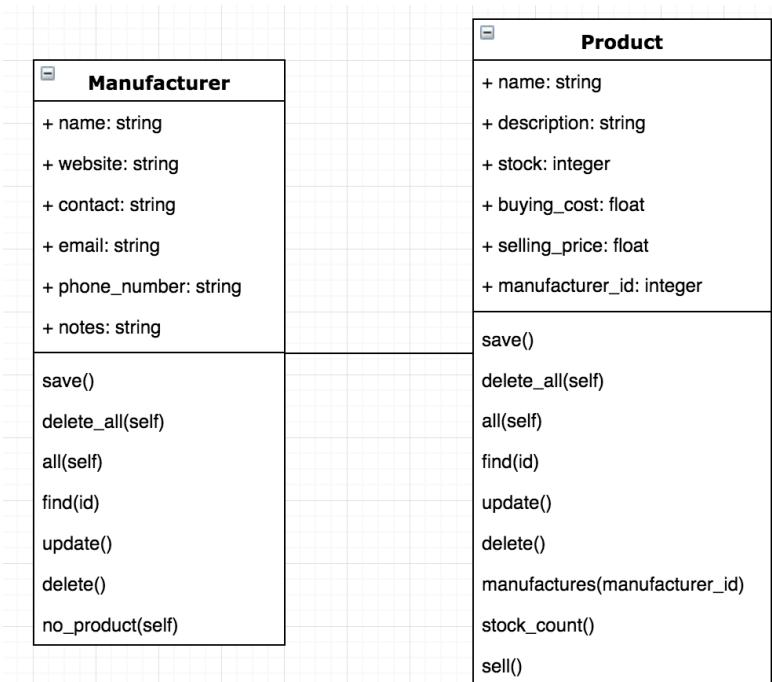
Unit	Ref	Evidence	
A&D	A.D.1	A Use Case Diagram  <b>Description:</b> Use Case Diagram showing all the things that an employee can do on the stock tracker in order to interact with the shop.	

**Paste Screenshot here**



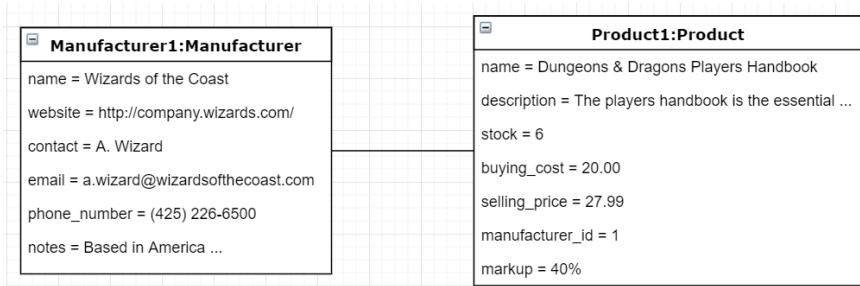
Unit	Ref	Evidence
A&D	A.D.2	<p>A Class Diagram</p> <p><b>Description:</b> Class Diagram showing two classes, manufacturer and products with their attributes, attribute types and methods.</p>

**Paste Screenshot here**



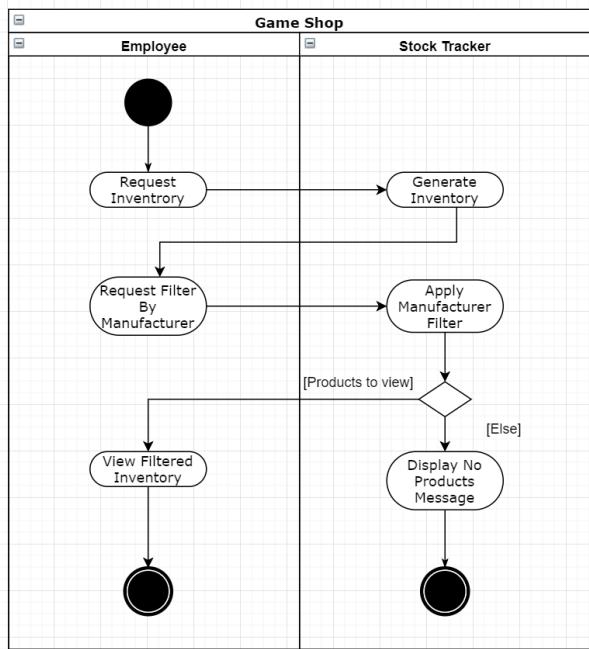
Unit	Ref	Evidence
A&D	A.D.3	<p>An Object Diagram</p> <p>Description: Object Diagram showing Manufacturer 1 and Product1 with dummy data and their interaction</p>

## Paste Screenshot here



Unit	Ref	Evidence	
A&D	A.D.4	<p>An Activity Diagram</p> <p><b>Description:</b> An Activity Diagram showing the process that both the employee and stock tracker go through when filtering the inventory including the decision point of whether there are any products to view for that manufacturer</p>	

Paste Screenshot here



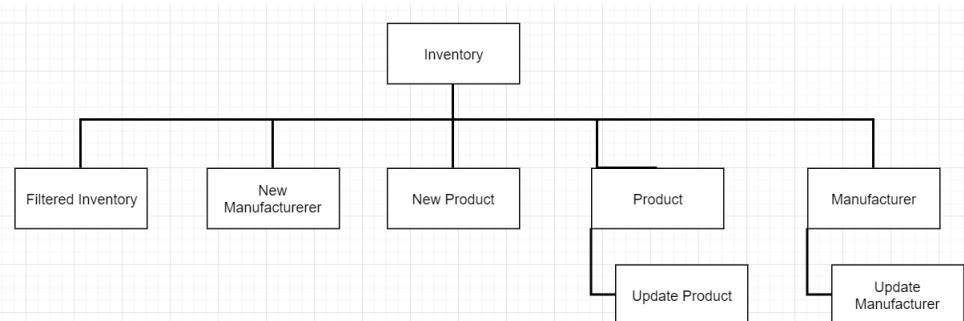
Unit	Ref	Evidence
A&D	A.D.6	<p>Produce an Implementations Constraints plan detailing the following factors:</p> <ul style="list-style-type: none"> <li>*Hardware and software platforms</li> <li>*Performance requirements</li> <li>*Persistent storage and transactions</li> <li>*Usability</li> <li>*Budgets</li> <li>*Time</li> </ul>
		<p><b>Description: Constraint diagram showing the few constraints that were on this project</b></p>

Paste Screenshot here

Topic	Possible Effect of Constraint on Product	Solution
Hardware and software platforms	Ruby, PSQL Sanatra. No easy way to have a dynamic front end.	Have a static front end that posts from various user inputs.
Performance requirements	Small shop so there is not a huge amount of data or functionality	Performance should not be a major factor
Persistent storage and transactions	psql was selected as the database for use	suitable for the data being used in this project
Usability	Should be usable for the stock side of a shop, not to be used by customers.	Ensure that planning takes into account the correct users for this software.
Budgets	No budget	Shouldn't be a problem for this project as there should be no expenses either.
Time	Project to be completed in under a week	Effective planning to ensure that realistic targets are set and use trello to keep track of tasks to do and completed.

Unit	Ref	Evidence
P	P.5	<p>User Site Map</p> <p><b>Description: User site map showing each page that a user can navigate to in a higherarchical structure</b></p>

Paste Screenshot here



Unit	Ref	Evidence	
P	P.6	<p>2 Wireframe Diagrams</p> <p><b>Description:</b> Two wire frames showing the layout of two pages from the site: the Inventory page and an individual manufacturer page</p>	

**Paste Screenshot here**

[Inventory](#) [New Product](#) [New Manufacturer](#)

Inventory

Product	Manufacturer	Stock	Price
<a href="#">Product Name</a>	<a href="#">Manufacturer Name</a>	Number	£xx.xx
<a href="#">Product Name</a>	<a href="#">Manufacturer Name</a>	Number	£xx.xx
<a href="#">Product Name</a>	<a href="#">Manufacturer Name</a>	Number	£xx.xx

Manufacturer With No Product

[Manufacturer Name](#)

[Manufacturer Name](#)

[Inventory](#) [New Product](#) [New Manufacturer](#)

Manufacturer

Name: Lorem ipsum dolor sit amet, consectetur

Website: [Linktoweb.com](#)

Contact: Lorem ipsum dolor sit amet, consectetur

Email: Lorem ipsum dolor sit amet, consectetur

Phone: Lorem ipsum dolor sit amet, consectetur

Notes: Lorem ipsum dolor sit amet, consectetur adipisciing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa

[Update Manufacturer](#)
[Delete Manufacturer](#)

Unit	Ref	Evidence	
P	P.10	Example of Pseudocode used for a method	
		<b>Description: Pseudocode explaining the function which returns an array of manufacturers which have no products associated with them</b>	

**Paste Screenshot here**

```
def self.no_product()
  sql = "SELECT manufacturers.* FROM manufacturers LEFT JOIN products ON
manufacturers.id = products.manufacturer_id WHERE products.id IS NULL"
  manufacturers = SqlRunner.run(sql)
  return manufacturers.map { |manufacturer| Manufacturer.new(manufacturer)}
end
```

```
# function to find manufacturers with no products and display them in an array
# the function should look in the database and compare the manufacturer and product tables
# the function should return only manufacturers from the database which have no products associated with them
# the function should then make each of these manufacturers returned into new manufacturer objects
# the manufacturer objects should be put into an array
# finally the function should return the manufacturer object array
```

Unit	Ref	Evidence
P	P.13	Show user input being processed according to design requirements. Take a screenshot of: * The user inputting something into your program * The user input being saved or used in some way
		<b>Description:</b> User inputting the manufacturer they want to filter by and the filter results

Paste Screenshot here



The screenshot shows a web browser window titled "Adventurer's Inventory". The URL bar shows "localhost:4567/inventory". The main content area has a red header bar with the title "Adventurer's Inventory" and navigation links for "Inventory", "New Product", and "New Manufacturer". Below this is a section titled "Inventory". A dropdown menu is open over a table, with "DnDice" selected. The table lists products with columns: Product, Manufacturer, Stock, and Price. The products listed are: Dungeons & Dragon (Wizards of the Coast, 6, £27.99), Magic The Gathering Core 2019 Deck Builders Toolkit (Wizards of the Coast, 0, £19.99), Prismatic Chromatic Dragon Dice Set (DnDice, 1, £29.95), Tokaido (Fun Forge, 10, £34.99), and Dungeons & Dragons Dungeon Masters Guide (Wizards of the Coast, 8, £27.99).

## Inventory

Filter By Manufacturer: Wizards of the Coast				Filter
	Product	Manufacturer	Stock	Price
Dungeons & Dragon	<a href="#">Dungeons &amp; Dragon</a>	Wizards of the Coast	6	£27.99
Chessex	<a href="#">Chessex</a>			
Magic The Gathering Core 2019 Deck Builders Toolkit	<a href="#">Magic The Gathering Core 2019 Deck Builders Toolkit</a>	Wizards of the Coast	0	£19.99
Prismatic Chromatic Dragon Dice Set	<a href="#">Prismatic Chromatic Dragon Dice Set</a>	DnDice	1	£29.95
Tokaido	<a href="#">Tokaido</a>	Fun Forge	10	£34.99
Dungeons & Dragons Dungeon Masters Guide	<a href="#">Dungeons &amp; Dragons Dungeon Masters Guide</a>	Wizards of the Coast	8	£27.99

## Manufacturers Without Products

[Chessex](#)  
[Fantasy Flight Games](#)




The screenshot shows a web browser window titled "Adventurer's Inventory". The URL bar shows "localhost:4567/inventory". The main content area has a red header bar with the title "Adventurer's Inventory" and navigation links for "Inventory", "New Product", and "New Manufacturer". Below this is a section titled "Inventory". A dropdown menu is open over a table, with "DnDice" selected. The table lists products with columns: Product, Manufacturer, Stock, and Price. The products listed are: Dungeons & Dragon (Wizards of the Coast, 6, £27.99), Magic The Gathering Core 2019 Deck Builders Toolkit (Wizards of the Coast, 0, £19.99), Prismatic Chromatic Dragon Dice Set (DnDice, 1, £29.95), Tokaido (Fun Forge, 10, £34.99), and Dungeons & Dragons Dungeon Masters Guide (Wizards of the Coast, 8, £27.99).

## Inventory

Filter By Manufacturer: DnDice				Filter
	Product	Manufacturer	Stock	Price
Dungeons & Dragons Player's Handbook	<a href="#">Dungeons &amp; Dragons Player's Handbook</a>	Wizards of the Coast	6	£27.99
Magic The Gathering Core 2019 Deck Builders Toolkit	<a href="#">Magic The Gathering Core 2019 Deck Builders Toolkit</a>	Wizards of the Coast	0	£19.99
Prismatic Chromatic Dragon Dice Set	<a href="#">Prismatic Chromatic Dragon Dice Set</a>	DnDice	1	£29.95
Tokaido	<a href="#">Tokaido</a>	Fun Forge	10	£34.99
Dungeons & Dragons Dungeon Masters Guide	<a href="#">Dungeons &amp; Dragons Dungeon Masters Guide</a>	Wizards of the Coast	8	£27.99

## Manufacturers Without Products

[Chessex](#)  
[Fantasy Flight Games](#)



Adventurer's Inventory

localhost:4567/inventory

# Adventurer's Inventory

Inventory   New Product   New Manufacturer

## Inventory DnDice

Product	Manufacturer	Stock	Price
<a href="#">Prismatic Chromatic Dragon Dice Set</a>	DnDice	2	£29.95



Unit	Ref	Evidence	
P	P.14	Show an interaction with data persistence. Take a screenshot of: * Data being inputted into your program * Confirmation of the data being saved	
		<b>Description:</b> The product page prior to alterations, the data being input into the update page and the product page confirming the updated data has been saved.	

**Paste Screenshot here**



**Adventurer's Inventory**

Inventory New Product New Manufacturer

## Product

Name: Dungeons & Dragons Player's Handbook

Description: The player's handbook is the essential reference for every Dungeons & Dragons roleplayer. It contains rules for character creation and advancement, battlegrounds and skills, exploration and combat, equipment, spells, and much more.

Stock: 6

Buying Cost: £20.00

Selling Price: £27.99

Markup: 40%

[Update Product](#) [Delete Product](#)



**Adventurer's Inventory**

Inventory New Product New Manufacturer

## Edit Product

Name: Dungeons & Dragons Player's Handbook

Description: The player's handbook is the essential reference for every Dungeons & Dragons roleplayer. It contains rules for character creation and advancement, battlegrounds and skills, exploration and combat, equipment, spells, and much more. I would recommend this book

Stock: 7

Buying Cost: £20.00

Selling Price: £27.99

Select A Manufacturer: Wizards of the Coast

[Update](#)

Adventurer's Inventory

localhost:4567/inventory/product/1

## Product

Name: Dungeons & Dragons Player's Handbook

Description: The player's handbook is the essential reference for every Dungeons & Dragons roleplayer. It contains rules for character creation and advancement, battlefields and skills, exploration and combat, equipment, spells, and much more. I would recommend this book.

Stock: 7

Buying Cost: 20.00

Selling Price: 27.99

Markup: 40%

[Update Product](#) [Delete Product](#)



Unit	Ref	Evidence
P	P.15	Show the correct output of results and feedback to user. Take a screenshot of: <ul style="list-style-type: none"> <li>* The user requesting information or an action to be performed</li> <li>* The user request being processed correctly and demonstrated in the program</li> </ul>
		<b>Description:</b> The inventory showing the manufacturer to be deleted, the user clicks on the manufacturer bringing up their page, the user clicks the delete button and then we see that the inventory no longer contains that manufacturer

Paste Screenshot here

Adventurer's Inventory

localhost:4567/inventory

## Inventory

Filter By Manufacturer: Wizards of the Coast [Filter](#)

Product	Manufacturer	Stock	Price
Magic The Gathering Core 2019 Deck Builders Toolkit	Wizards of the Coast	0	£19.99
Prismatic Chromatic Dragon Dice Set	DnDice	2	£29.95
Tokaido	Fun Forge	10	£34.99
Dungeons & Dragons Dungeon Masters Guide	Wizards of the Coast	8	£27.99
Dungeons & Dragons Player's Handbook	Wizards of the Coast	7	£27.99

### Manufacturers Without Products

[Chessex](#)  
[Fantasy Flight Games](#)

localhost:4567/inventory/manufacturer/5



Adventurer's Inventory

localhost:4567/inventory/manufacturer/5

## Manufacturer

Name: Chessex

Website: <http://www.chessex.com/>

Contact Name: C.Dicey

Email: c.dicey@chessex.com

Phone: 0800 000 000

Notes: Largest dice manufacturer in the UK

[Update Manufacturer](#) [Delete Manufacturer](#)



Adventurer's Inventory

localhost:4567/inventory

## Inventory

Filter By Manufacturer: Wizards of the Coast ▾ [Filter](#)

Product	Manufacturer	Stock	Price
<a href="#">Magic The Gathering Core 2019 Deck Builders Toolkit</a>	<a href="#">Wizards of the Coast</a>	0	£19.99
<a href="#">Prismatic Chromatic Dragon Dice Set</a>	<a href="#">DnDice</a>	2	£29.95
<a href="#">Tokaido</a>	<a href="#">Fun Forge</a>	10	£34.99
<a href="#">Dungeons &amp; Dragons Dungeon Masters Guide</a>	<a href="#">Wizards of the Coast</a>	8	£27.99
<a href="#">Dungeons &amp; Dragons Player's Handbook</a>	<a href="#">Wizards of the Coast</a>	7	£27.99

### Manufacturers Without Products

[Fantasy Flight Games](#)



Unit	Ref	Evidence	
P	P.11	Take a screenshot of one of your projects where you have worked alone and attach the Github link.	
		Description: <a href="https://github.com/anne-other/game_shop_ruby_project">https://github.com/anne-other/game_shop_ruby_project</a>	

Paste Screenshot here

The screenshot shows a web application titled "Adventurer's Inventory". The header includes a logo, the title, and navigation links for "Inventory", "New Product", and "New Manufacturer". The main section is titled "Inventory" and displays a table of products. A filter dropdown is set to "Wizards of the Coast" and a "Filter" button is visible. The table columns are "Product", "Manufacturer", "Stock", and "Price". The data includes:

Product	Manufacturer	Stock	Price
Dungeons & Dragons Player's Handbook	Wizards of the Coast	6	£27.99
Magic The Gathering Core 2019 Deck Builders Toolkit	Wizards of the Coast	0	£19.99
Prismatic Chromatic Dragon Dice Set	DnDice	3	£29.95
Tokaido	Fun Forge	10	£34.99
Dungeons & Dragons Dungeon Masters Guide	Wizards of the Coast	8	£27.99

Below the table, there is a section titled "Manufacturers Without Products" with two entries: "Chessex" and "Fantasy Flight Games". A large, semi-transparent red silhouette of a person's head and shoulders is overlaid on the right side of the page.

Unit	Ref	Evidence	
P	P.12	Take screenshots or photos of your planning and the different stages of development to show changes.	
		Description:	

Paste Screenshot here

The screenshot shows a character creation or planning interface. At the top left is a portrait of a woman with the name "Fred" below it. To the right is a section for "Your proto-persona's name" with a placeholder "Basic characteristics go here" and a "Skills" section. Below these are "Behaviours" and "Demographic Information" sections. The "Behaviours" section lists items like "Is here on game night", "Spends free time gaming", and "Watches online and other RPG streams". The "Demographic Information" section includes "35 years old", "Sleeps well", "Marrried with 2 kids", and "Fairly well off".

At the bottom left is a "Demographics" section with a list of activities such as "Takes bus to work", "Has a long-term girl friend", and "Lives with 3 flatmates who are also similar things". To the right is a "Needs and goals" section with items like "Needs to make more game money", "Has positive interaction with customers", and "Create new stock input weekly". Another "Needs and goals" section is at the bottom right.

As a...	I want to...	So that...
Geeek culture fan	Know all about the games in store	I can have positive interactions with customers
Student	be able to sell to customers	I can have beer and game money
Part time employee	have quick access to stock information	I can serve customers
Part time employee	have an easy way to input new stock	my job is made easier
Person who games socially	know all about the latest products	I can show off to my friends
Employee	find the price of products quickly	I can serve customers
A keen gamer	be able to update games	The information is correct

As a...	I want to...	So that...
Mum	understand the games my shop sells a little	I don't give my kids something inappropriate
Shop owner	keep track of how much stock I have	I can order more stock where required
Bussiness owner	keep track of which manufacturer's products we have	I can maintain good relationships with those manufacturers
Provider for my family	keep track of the buying and selling price of games	to ensure my bussiness is making money
Good networker	be able to add or delete manufacturers	My shop is up to date with the changing relationships with different manufacturers
Someone with an eye for nich markets	be able to add or delete a game	My shop is selling the most popular games

User action(s)	Step 1	Step 2	Step 3	User action(s)	Step 1	Step 2	Step 3	Step 4	Step 5
User action(s)	Fred goes onto the inventory and scans down the stock column to see what the stock levels are like	Fred sees whether they have stock of a particular game	Fred finds the form is prepopulated with the game information and edits any appropriate sections	User action(s)	Fred goes onto the stock site to find a game he wants to find information on	Fred clicks on a game he is interested in	Fred reads the description of the game	Fred finds that the form is prepopulated with the game information and edits any appropriate sections	Fred clicks a button to submit the changes
System response	Database of all games, inventory shows all items including a method to show the number of stock	A method to show on the inventory if something is out of stock	Prepopulate the form with values from the database. Form that gets the information required to save the changes to the database table	System response	Database of all games, inventory shows all items	Links available in the index which will take the user to find further information about the game	Method to find one instance of a game from the stock database and displays more information about that game including a game description	Post method to give the information from the form to the database	Return to inventory page

User action(s)	Step 1	Step 2	Step 3	User action(s)	Step 1	Step 2	Step 3	Step 4	User action(s)	Step 1	Step 2	Step 3	Step 4
User action(s)	Shiona goes onto the inventory and looks at the manufacturer column	Shiona clicks on a manufacturer to find out the details associated with that manufacturer	Shiona will look at the information on the individual manufacturers page	User action(s)	Shiona goes onto the inventory and finds the manufacturer she wants to delete	Shiona clicks on a button to delete the manufacturer	Shiona is returned to the inventory page where manufacturer will no longer be present	Shiona is returned to the inventory page where the new manufacturer will be displayed	User action(s)	Shiona fills out the form to add a new manufacturer	Shiona clicks a button to submit the new manufacturer	Shiona is returned to the inventory page where the new manufacturer will be displayed	Shiona is returned to the inventory page
System response	Database of all manufacturers, inventory shows all manufacturers in a column next to each game	Inventory table should include a link for each manufacturer which will take you to an individual manufacturer table	Database of manufacturers, inventory shows all relevant information for one particular manufacturer on this page	System response	Database of all games, inventory shows all items and manufacturers	Link on the inventory for each manufacturer which takes the user to the individual manufacturer page	Button calls delete function on the database, ensure that the database has a cascade on delete so the games are removed at the same time	Return to inventory page	System response	Database of all games, inventory shows all items and manufacturers	Form that gets the information required to save a new manufacturer to the database table	Post method to give the information from the form to the database	Return to inventory page

Inventory

Product	Manufacturer	Stock	Price	Sell
Dungeons & Dragons	Wizards of the Coast	15	\$27.99	Sell
Player's Handbook				
Magic the Gathering Core Set	Wizards of the Coast	0	\$19.95	Sell
Canada: D&D Player's Guide	D&D	2	\$29.95	Sell

Manufacturers with no product

- Fire Forge

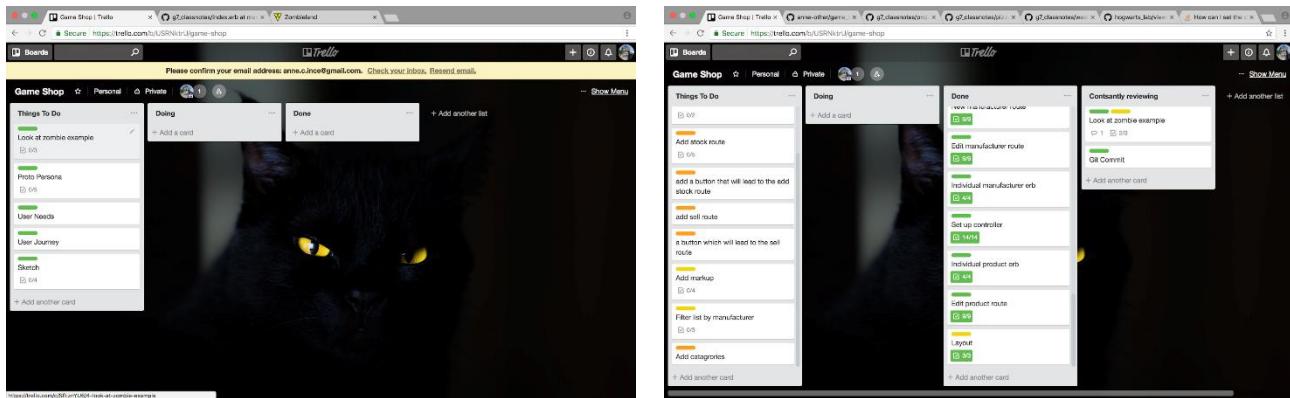
Inventory

Name	Website	Contact	Email	Phone	Note
Wizards of the Coast	www.wizards.com	Customer Support	customer.wotc@wizards.com	1800 200 000	Never forget to ask for

Update Manufacturer Delete Manufacturer

New Product

Name	Description	Stock	Buying Cost \$	Selling Cost \$	Manufacturer	Add
						Add



## Week 7

Unit	Ref	Evidence
P	P.16	Show an API being used within your program. Take a screenshot of: * The code that uses or implements the API * The API being used by the program whilst running
		<b>Description:</b> The get equipment function passes the url of the API into the request function which uses promises and fetch to return the data about the equipment from the API, the get equipment function then publishes the data from the API to the view where it is rendered into the browser.

### Paste Screenshot here

```
Equipment.prototype.getEquipment = function () {
  const request = new Request("http://www.dnd5eapi.co/api/equipment")
  request.get().then((data) => {
    this.equipment = data.results;
    PubSub.publish('Equipment:equipment-data-loaded', this.equipment);
  }).catch((error) => {
    console.error(error);
  })
};
```

```
const Request = function (url) {
  this.url = url
}

Request.prototype.get = function () {
  return fetch(this.url)
  .then(response => response.json());
};

module.exports = Request;
```

D&D 5th Edition Equipment

All Categories

**Club**

- Category: Weapon
- Weight: 2
- Cost: 1 sp
- Weapon Category: Simple
- Range: Melee
- Damage: 1d4 Bludgeoning

**Dagger**

- Category: Weapon
- Weight: 1
- Cost: 2 gp
- Weapon Category: Simple
- Range: Melee
- Damage: 1d4 Piercing

**Greatclub**

- Category: Weapon
- Weight: 10

Unit	Ref	Evidence
P	P.18	Demonstrate testing in your program. Take screenshots of: * Example of test code * The test code failing to pass * Example of the test code once errors have been corrected * The test code passing
		<b>Description:</b> Test fails because highest card has not been defined for self, after making it a self method the test passes.

Paste Screenshot here

```
def test_highest_card()
    result =
    CardGame.highest_card(@card1,
        @card2)
    assert_equal(@card1, result)
end
```

```
|→ pda_static_and_dynamic_testing_tasks git:(master) ✘ ruby specs/card_spec.rb
Run options: --seed 54971

# Running:

E

Finished in 0.001124s, 889.6797 runs/s, 0.0000 assertions/s.

1) Error:
CardsTest#test_highest_card:
NoMethodError: undefined method `highest_card'
for CardGame:Class
    specs/card_spec.rb:25:in `test_highest_card'

1 runs, 0 assertions, 0 failures, 1 errors, 0 skips
```

```

def self.highest_card(card1,
card2)
  if card1.value > card2.value
    return card1
  else
    card2
  end
end

```

```

↳ pda_static_and_dynamic_testing_tasks git:(master) ✘ ruby specs/card_spec.rb
Run options: --seed 46341

# Running:

.

Finished in 0.000896s, 1116.0714 runs/s, 1116.0
714 assertions/s.

1 runs, 1 assertions, 0 failures, 0 errors, 0 s
kips

```

## Week 9

Unit	Ref	Evidence	
P	P.1	Take a screenshot of the contributor's page on Github from your group project to show the team you worked with.	
		<b>Description: Contributors to the project from git hub</b>	

### Paste Screenshot here



Unit	Ref	Evidence	
P	P.2	<p>Take a screenshot of the project brief from your group project.</p> <p><b>Description: Project brief</b></p>	

Paste Screenshot here

## Educational App

---

The BBC are looking to improve their online offering of educational content by developing some interactive browser applications that display information in a fun and interesting way. Your task is to make an a Minimum Viable Product or prototype to put forward to them - this may only be for a small set of information, and may only showcase some of the features to be included in the final app.

### MVP

---

A user should be able to:

- view some educational content on a particular topic
- be able to interact with the page to move through different sections of content

### Example Extensions

---

- Use an API to bring in content or a database to store information.
- Use charts or maps to display your information to the page.

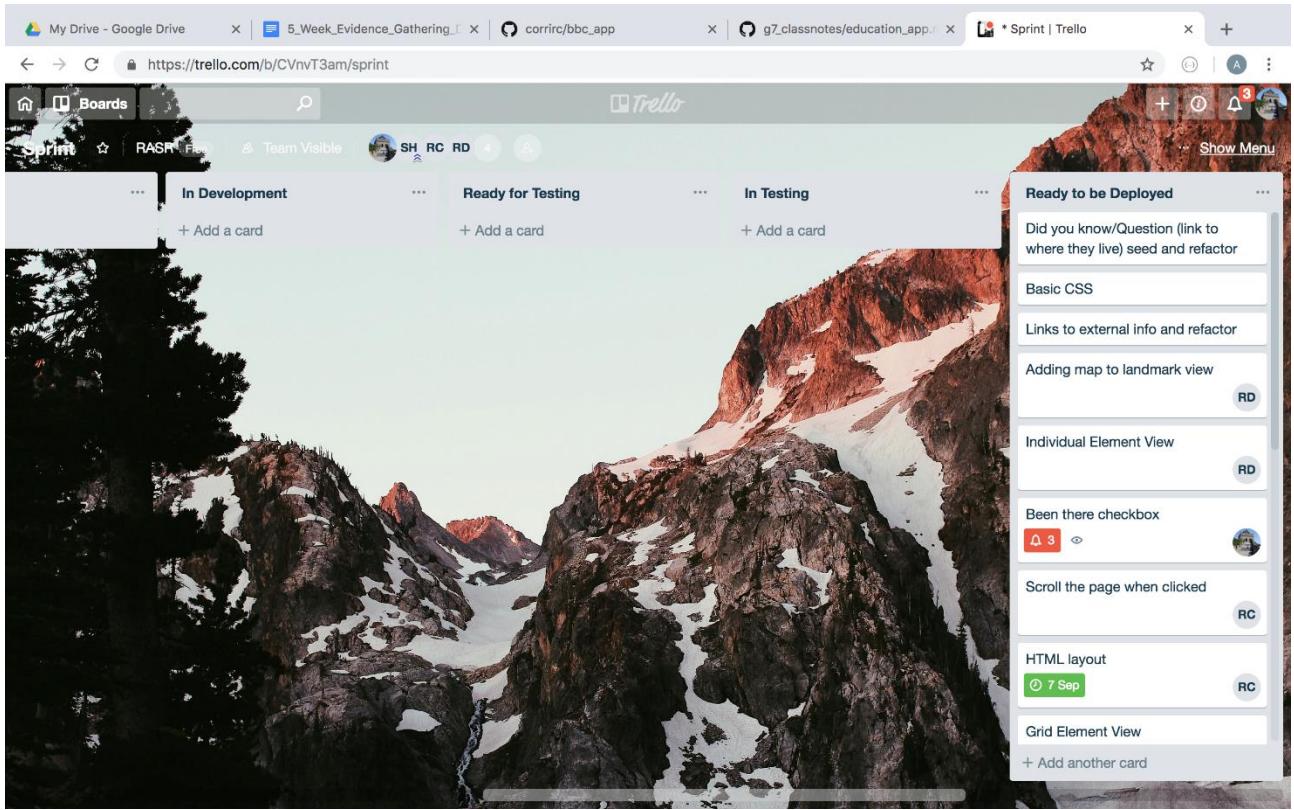
### API, Libraries, Resources

---

- <https://www.highcharts.com/> HighCharts is an open-source library for rendering responsive charts.
- <https://leafletjs.com/> Leaflet is an open-source library for rendering maps and map functionality.

Unit	Ref	Evidence	
P	P.3	Provide a screenshot of the planning you completed during your group project, e.g. Trello MOSCOW board.	
		<b>Description: Our group trello board on completion</b>	

**Paste Screenshot here**



Unit	Ref	Evidence	
P	P.4	Write an acceptance criteria and test plan.	

Paste Screenshot here

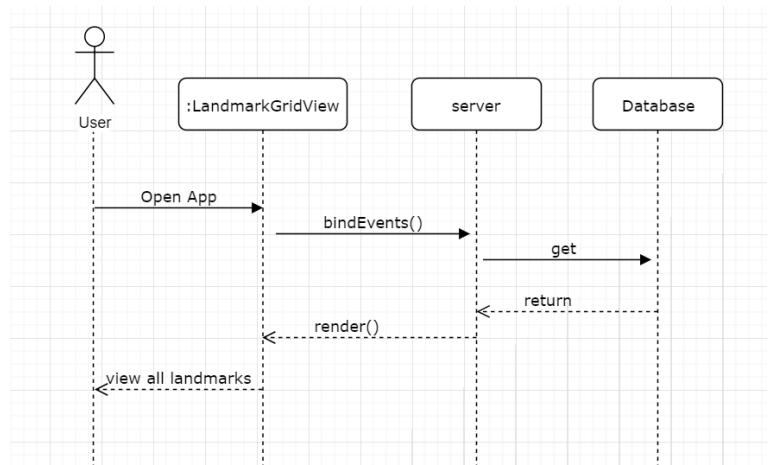
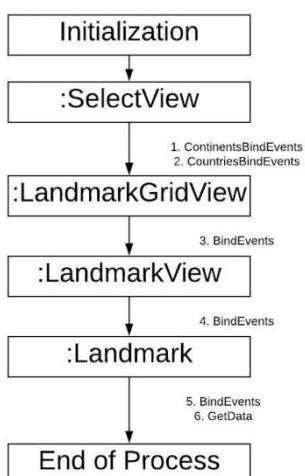
**Acceptance Criteria**

Acceptance Criteria	Expected Result/Output	Pass / Fail
Student is able to access a list of all landmarks	A list of all landmarks is displayed when URL is accessed	Pass
A student can click on a landmark to display all of its information	When student clicks, a view is populated with all the landmark information	Pass
<b>Extensions</b>		
A student can click on a link inside the landmark view to access external webpage about the landmark	When student clicks the "Get More Information" button, they are taken to an external URL (if landmark has its own page, otherwise a National Geographic or Lonely planet Link)	Pass
A student can mark a landmark as visited	Student can check a box toggle a switch to mark a landmark as one they have visited	Pass
Student can view a map of landmark location	When student clicks, a map is included in the landmark view as part of the information	Pass
A student can filter landmarks by set criteria	Students can pull down a list, which will be populated with information from seeds - e.g. countries, etc	Pass

Unit	Ref	Evidence	
P	P.7	Produce two system interaction diagrams (sequence and/or collaboration diagrams).	

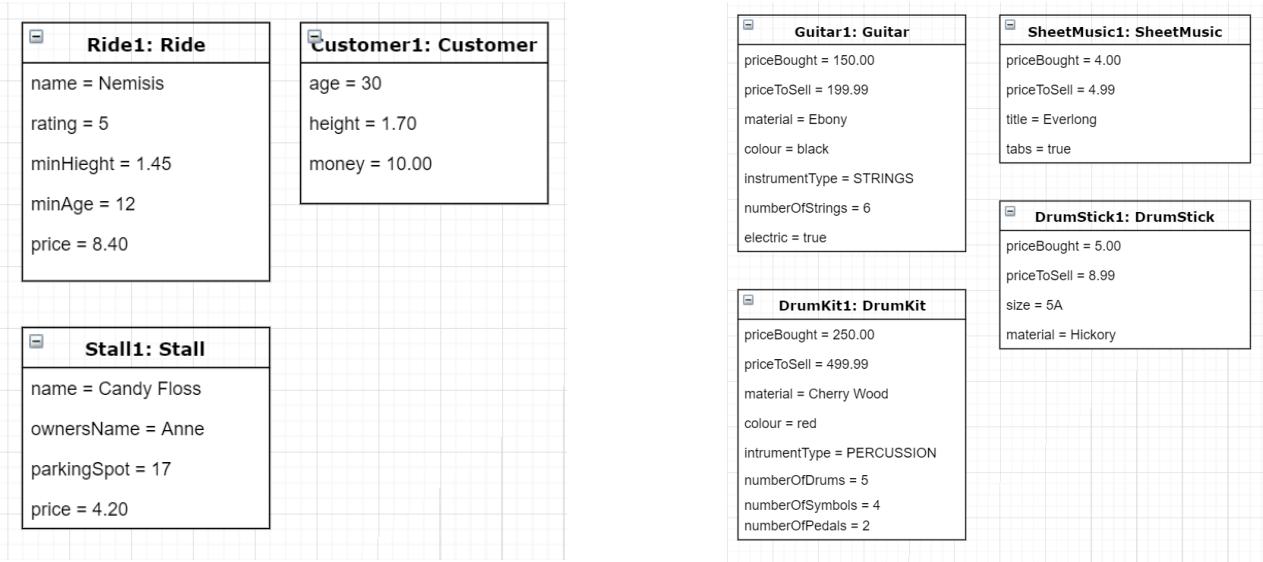
Paste Screenshot here

Collaboration diagram of a BBC educational landmark app



Unit	Ref	Evidence	
P	P.8	<p>Produce two object diagrams.</p> <p><b>Description:</b></p>	

### Paste Screenshot here



Unit	Ref	Evidence	
P	P.17	<p>Produce a bug tracking report</p> <p><b>Description:</b> bug tracking report</p>	

### Paste Screenshot here

#### Bug Tracking Report

Issue		Solution	Pass/ Fail
Database can load seeds	Fail	Fix syntax errors in seeds	Pass
Student can click on a landmark and page will scroll down to the information.	Fail	Fix placement of scroll method within the code to allow content to load fully before scroll occurs.	Pass
Student can click on link within landmark view to be taken to an external website.			Pass
Toggle switch is displayed on the grid item view for student to interact with	Fail	Added missing line of code in CSS file	Pass
Toggle switch will update the database with changes (true or false) as to whether student has been to the landmark	Fail	Fixed code so that it was updating a single parameter within the database rather than the whole object.	Pass
Landmark view will load map to show location of the selected landmark While it worked for some landmarks, it did not work for all (as bringing back local places "Sphinx Medical" or some not even found)	Fail	Add additional parameters to the method to get name, location and continent of landmark (Future: Add lat/lng to seeds?)	Pass

## Week 12

Unit	Ref	Evidence	
I&T	I.T.7	The use of Polymorphism in a program and what it is doing.	
		<b>Description:</b> Polymorphism is used here to ensure that anything that extends the abstract class StockItem can be added to the same array.	

Paste Screenshot here

```
public class Shop {
    ArrayList<StockItem> stockItems;

    public Shop(ArrayList<StockItem> stockItems) {
        this.stockItems = stockItems;
    }
}
```

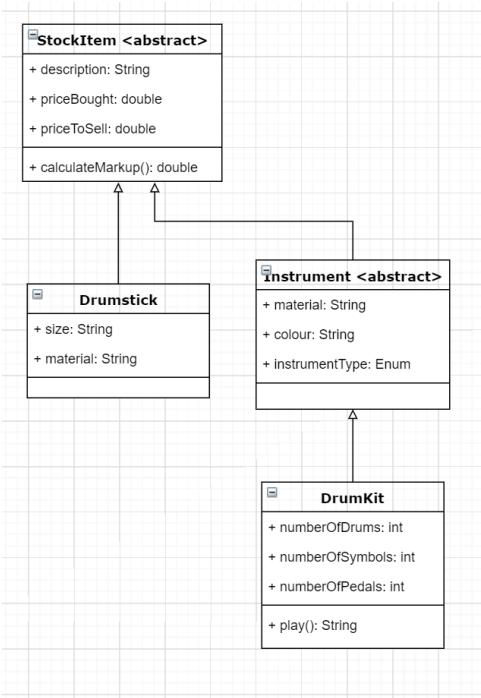
```
public abstract class StockItem {
    private String description;
    private double priceBought;
    private double priceToSell;

    public StockItem(String description, double priceBought, double priceToSell) {
        this.description = description;
        this.priceBought = priceBought;
        this.priceToSell = priceToSell;
    }
}
```

```
public class SheetMusic extends StockItem {
```

Unit	Ref	Evidence	
A&D	A.D.5	An Inheritance Diagram  <b>Description:</b> DrumKit inherits Instrument, Instrument and Drumstick inherit StockItem	

Paste Screenshot here



Unit	Ref	Evidence	
I&T	I.T.1	<p>The use of Encapsulation in a program and what it is doing.</p> <p><b>Description: Private fields with getters and no setters so data is protected</b></p>	

Paste Screenshot here

```
public class DrumKit extends Instrument implements Playable {

    private int numberOfDrums;
    private int numberOfSymbols;
    private int numberOfPedals;

    public DrumKit(String description, double priceBought, double priceToSell, String material,
                   String colour, Enum InstrumentType, int numberOfDrums, int numberOfSymbols,
                   int numberOfPedals) {
        super(description, priceBought, priceToSell, material, colour, InstrumentType);
        this.numberOfDrums = numberOfDrums;
        this.numberOfSymbols = numberOfSymbols;
        this.numberOfPedals = numberOfPedals;
    }

    public String play(){
        return "Boom";
    }

    public int getNumberOfDrums() {
        return numberOfDrums;
    }

    public int getNumberOfSymbols() {
        return numberOfSymbols;
    }
}
```

Unit	Ref	Evidence	
I&T	I.T.2	<p>Take a screenshot of the use of Inheritance in a program. Take screenshots of:</p> <ul style="list-style-type: none"> <li>*A Class</li> <li>*A Class that inherits from the previous class</li> <li>*An Object in the inherited class</li> <li>*A Method that uses the information inherited from another class.</li> </ul>	<p><b>Description: Instrument abstract class, Guitar inherits from Instrument Class, Guitar guitar is an object in the inherited class and getMaterial() is a method that is inherited from another class.</b></p>

### Paste Screenshot here

```
public abstract class Instrument extends StockItem {

    private String material;
    private String colour;
    private Enum InstrumentType;

    public Instrument(String description, double priceBought, double priceToSell, String material,
                     String colour, Enum InstrumentType) {
        super(description, priceBought, priceToSell);
        this.material = material;
        this.colour = colour;
        this.InstrumentType = InstrumentType;
    }

    public String getMaterial() {
        return material;
    }

    public String getColour() {
        return colour;
    }

    public Enum getInstrumentType() {
        return InstrumentType;
    }
}
```

```
public class Guitar extends Instrument implements Playable {

    private int numberOfStrings;
    private boolean electric;

    public Guitar(String description, double priceBought, double priceToSell, String material,
                 String colour, Enum InstrumentType, int numberOfStrings, boolean electric) {
        super(description, priceBought, priceToSell, material, colour, InstrumentType);
        this.numberOfStrings = numberOfStrings;
        this.electric = electric;
    }
}
```

```
Guitar guitar;

@Before
public void setUp() throws Exception {
    guitar = new Guitar( description: "Guitar", priceBought: 150.00, priceToSell: 199.99, material: "Ebony",
                        colour: "Black", InstrumentType.STRINGS, numberOfStrings: 6, electric: true);
}
```

```
@Test
public void hasmaterial() {
    assertEquals( expected: "Ebony", guitar.getMaterial());
}
```

Unit	Ref	Evidence	
P	P.9	Select two algorithms you have written (NOT the group project). Take a screenshot of each and write a short statement on why you have chosen to use those algorithms.	
		<b>Description:</b> first algorithm loops through the stock items to find the relevant item and then removes the item from stock, the second also loops though the stock but then adds each item mark-up to the total profit.	

Paste Screenshot here

```
public void removeItemFromStock(ISell itemToBeSold){
    ISell foundItem = null;
    for (ISell item : this.stock){
        if (item == itemToBeSold){
            foundItem = item;
        }
    } this.stock.remove(foundItem);
}
```

```
public Double calculateTotalPossibleProfit(){
    Double profit = 0.00;
    for (ISell item : this.stock){
        Double itemprofit = item.markup();
        profit += itemprofit;
    }
    return profit;
}
```