

REF I.T 5

Input

```
33 # I.T 5
34 array_of_colours = ["Red", "Green", "Blue"]
35
36 def capitalise_colours(array_of_colours)
37   array_of_colours_capitalize = []
38   array_of_colours.each {|colour| array_of_colours_capitalize << colour.upcase}
39 end
40
41 puts capitalise_colours(array_of_colours)
42 print capitalise_colours(array_of_colours)
```

Output

```
[→ PDA ruby evidence_log.rb
RED
GREEN
BLUE
["RED", "GREEN", "BLUE"]
```

REF I.T 6

Input

```
11 # I.T 6
12 dog_hash = {:Name => "Vince", :Breed => "Labrador", :Colour => "Black", :age => "3"}
13
14 def convert_dog_age_to_integer(dog_hash)
15   return dog_hash[:age].to_i
16 end
17
18 puts convert_dog_age_to_integer(dog_hash)
```

Output

```
[→ PDA ruby evidence_log.rb
3
```

REF I.T 4

Input

```
23 # I.T 4
24 array_of_colours = ["Red", "Green", "Blue"]
25
26 def sort_colours_in_alphabetic(array_of_colours)
27   array_of_colours.sort
28 end
29
30 puts sort_colours_in_alphabetic(array_of_colours)
```

Output

..clan_work/PDA

```
[→ PDA ruby evidence_log.rb  
Blue  
Green  
Red
```

REF I.T 3

Input

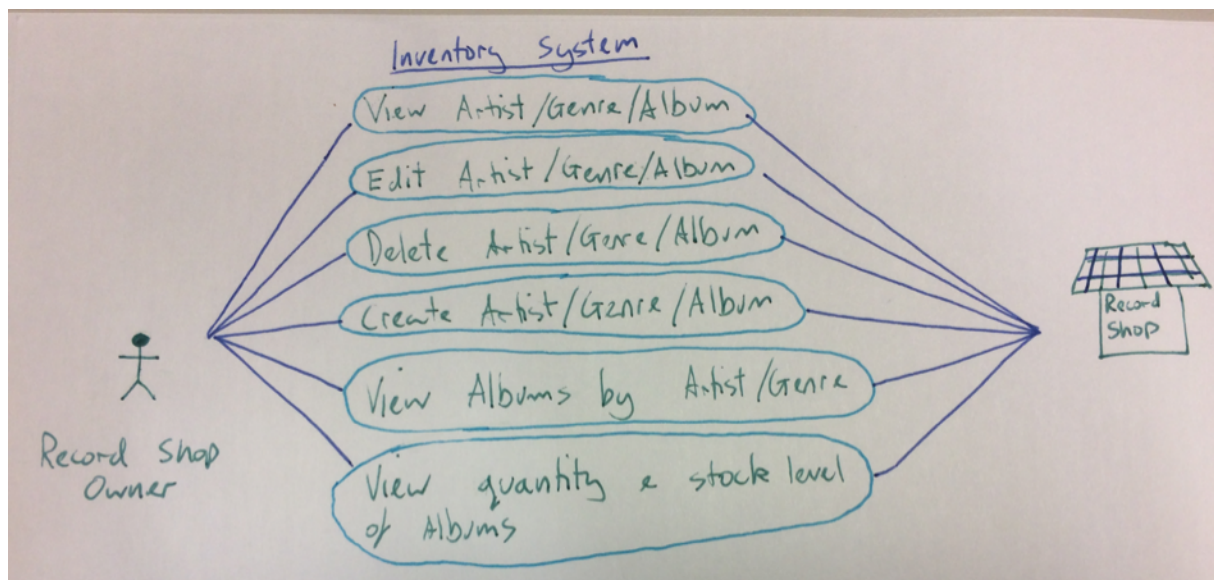
```
1 # I.T.3  
2 dog_hash = {:Name => "Vince", :Breed => "Labrador", :Colour => "Black", :age => "3"}  
3  
4 def find_dog_name (dog_hash)  
5   return dog_hash[:Name]  
6 end  
7  
8 puts find_dog_name(dog_hash)
```

Output

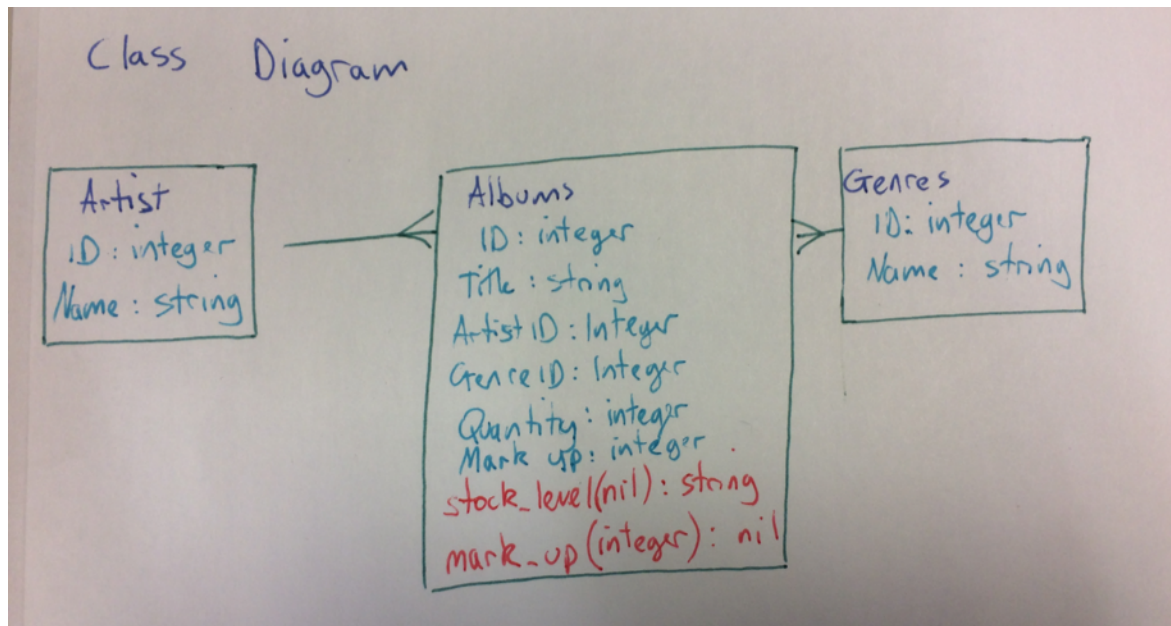
ruby

```
[→ PDA ruby evidence_log.rb  
Vince
```

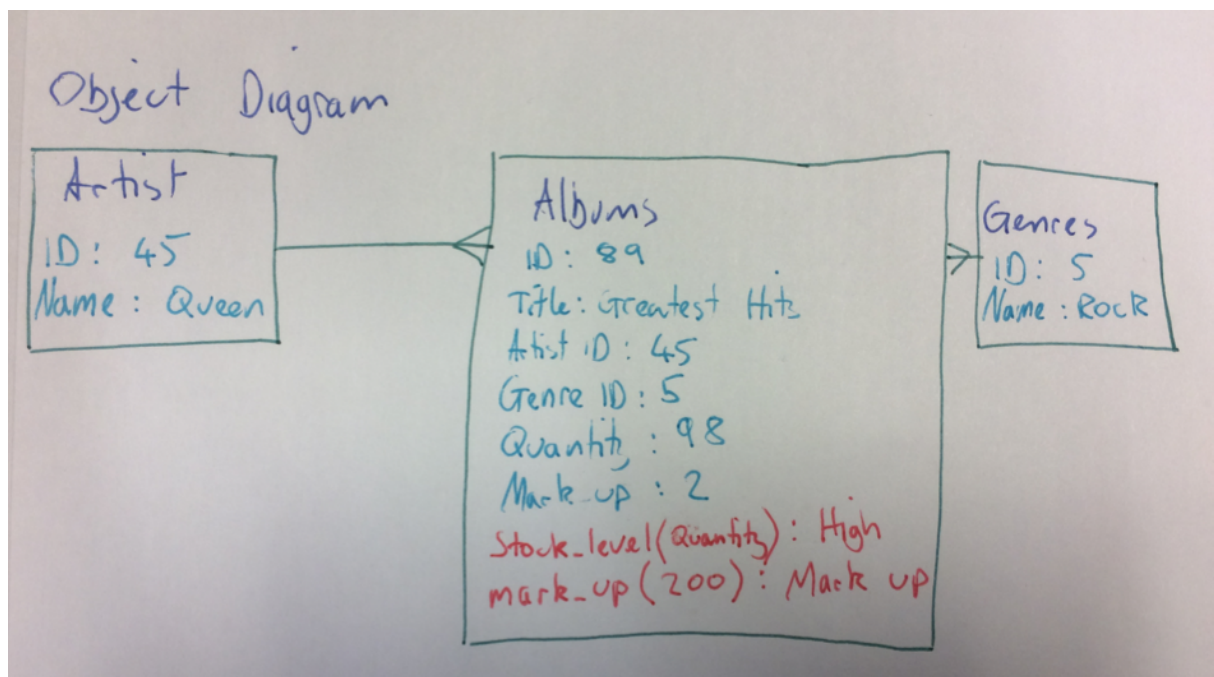
REF A.D 1



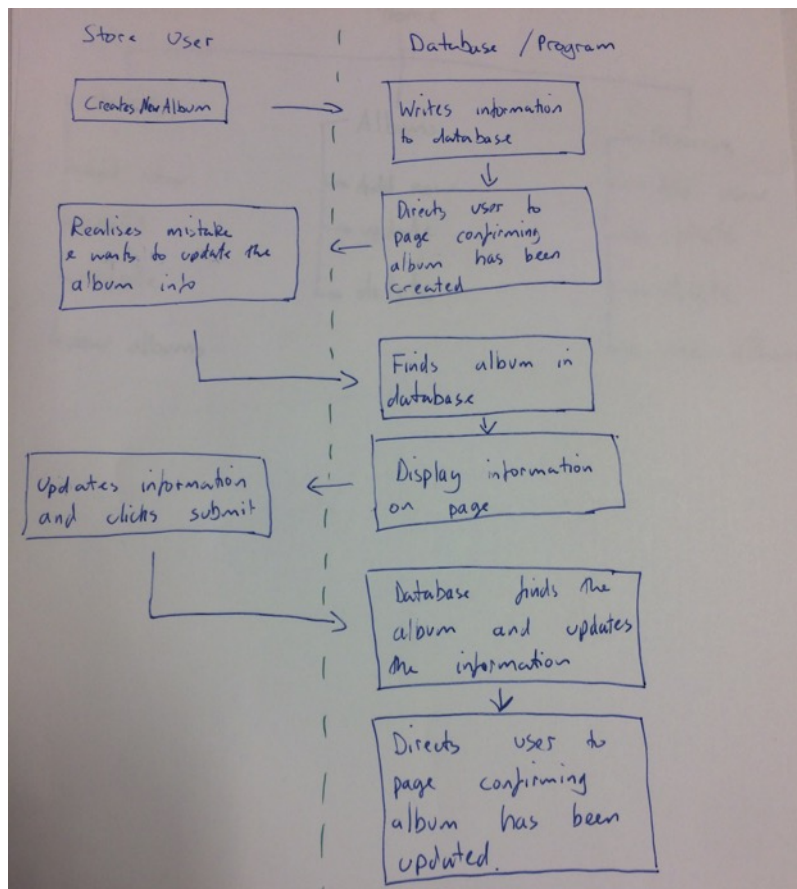
REF A.D 2



REF A.D 3



REF A.D 4

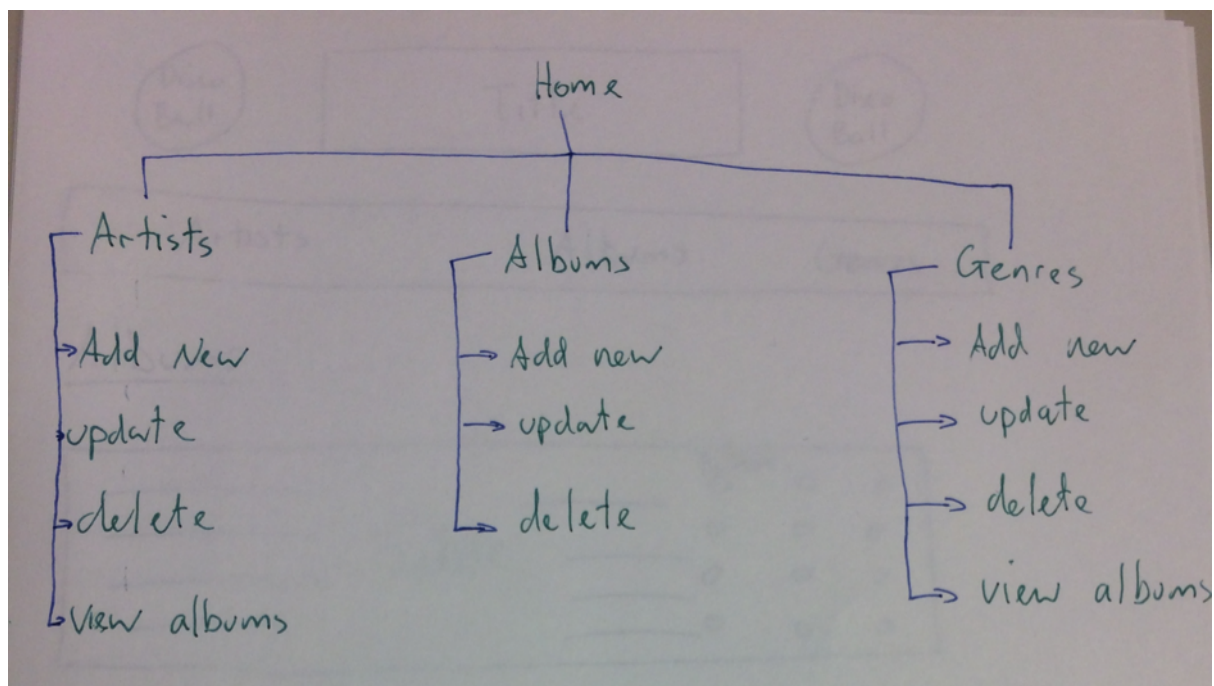


REF A.D 6

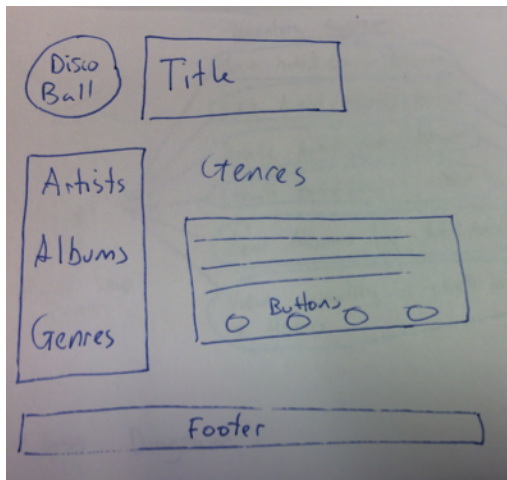
Topic	Possible Effect of Constraint on Product	Solution
Hardware and software platforms	A server which cannot hold all the information which means the shop wont be able to store all their album information. This will affect the shops performance.	A server which is big enough to hold the data and enough performance to deal with the transactions from the required amount of users.
Performance requirements	Only applicable to one shop. This will be an issue if more shops need to be opened and means the system is redundant in current form.	Create a class for shops which allows multiple shops to have their own inventory
Persistent storage and transactions	The database will run slower as more albums are added. This will make the user experience poor and may drive to the system not being used.	Not an issue at the moment but would require more efficient code if being scaled up.

Usability	Site map too complicated and user gets lost in the program. Purpose of the system is to aid the user and there is potential for it to be a hindrance.	Carry out UX testing and minimise the amount of steps to complete a task.
Budgets	Scope creep could lead to more time being required to complete the project.	Ensure the project scope is maintained and achievable from day 1
Time limitations	All functionality partially complete (ie 5 functions at 75%) which means whole program is unusable.	Break functions down and ensure they are working before moving on to next function. Allows some functionality to be useable eg (4 at 100% and 1 at 25%).

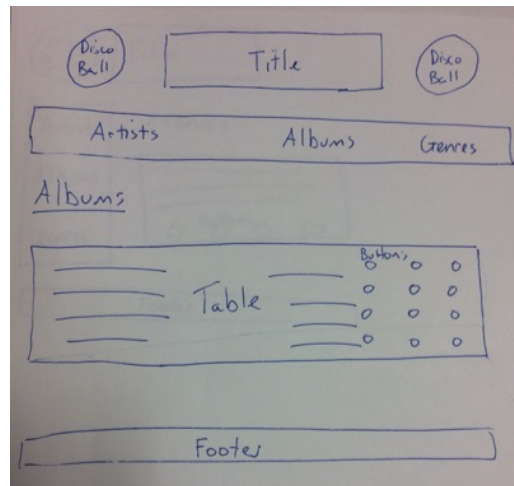
Ref P 5



Ref P 6



Example1



Example 2

Ref P 10

```
# 1. A function which finds an album based on its artist id on the album table
# 2. sql code which searches the database for all the information related
# to a specified artist_id
# 3. the output of the sql code is held as an array of multiple albums
# 4. the array of albums are mapped to a new array and each new entry
# is passed through as a new class to ensure all class instances are created
```

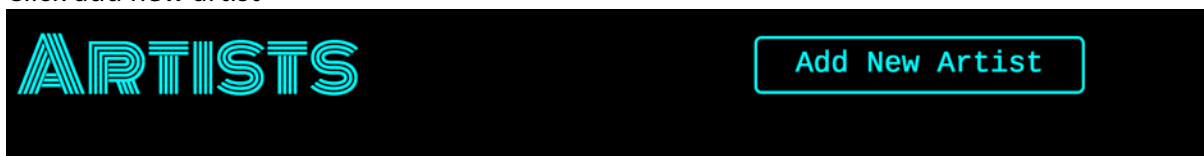
```
def self.albums_by_artist(id)

  sql = "SELECT * FROM albums WHERE artist_id = $1"

  values = [id]
  albums = SqlRunner.run(sql, values)
  return albums.map{ |album| Album.new( album ) }
end
```

Ref P13

Click add new artist



Add new artist page opens

ADD NEW ARTISTS

Artist Name:

Add Artist

Ref P14

Input album information

UPDATE ALBUM



Album Title:

Favourite worst nightma

Artist Name:

Arctic Monkeys

Genre:

Rock

Gross Cost:

£ 5.87

Album URL:

https://s1-ssl.mzstatic.c

Quantity:

21

Mark Up:

133

UPDATE

Album information updated in dashboard.

ALBUMS

Add New Album

Album Title	Artist Name	Quantity	Stock Level	Last Bought	View	Edit	Delete
The getaway	The Red Hot Chili Peppers	2	Low				
Am	Arctic Monkeys	9	Medium				
Favourite worst nightmare	Arctic Monkeys	21	High				

Ref P15

Click delete classical button

Genre Name	View Related Albums	Edit Genre	Delete Genre
Rock			
Pop			
Jazz			
Classical			
Metal			

Classical entry has been removed

Genre Name	View Related Albums	Edit Genre	Delete Genre
Rock			
Pop			
Jazz			
Metal			