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.

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

```
def setup()
  @song1 = Song.new("Monkey Wrench", "Foo Fighters")
  @song2 = Song.new("Song 2", "Blur")
  @song3 = Song.new("Hey Jude", "The Beatles")
  @song4 = Song.new("Sweet Caroline", "Neil Diamond")
  @song5 = Song.new("Encore", "Jay-Z")
  @songs = [@song1, @song2, @song3, @song4]
```

```
def song()
  return @songs.count
end

def add_song(song)
  @songs.push(song)
end
```

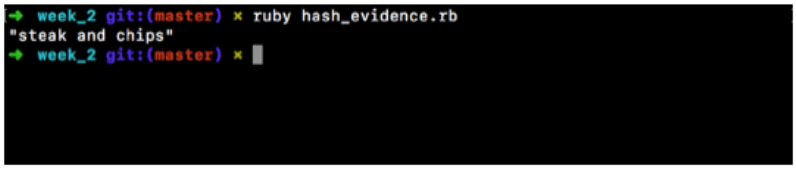
```
def test_room_has_songs__added()
  @room.add_song(@song2)
  assert_equal(1, @room.songs)
end
```

```
➜ Caraoke_Homework git:(master) ✗ ruby specs/room_spec.rb
Run options: --seed 20587

# Running:

.

Finished in 0.000973s, 1027.7492 runs/s, 1027.7492 assertions/s.

1 runs, 1 assertions, 0 failures, 0 errors, 0 skips
➜ Caraoke_Homework git:(master) ✗
```

This screenshot shows an array of songs, with a function that adds a song to the array and the test for this, with a screenshot of the test passing in the terminal.

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

```ruby
diners = {sarah: {starter: "soup",
        main: "steak and chips",
        dessert: "lemon tart"},

        graeme: {starter: "bruschetta",
        main: "spaghetti",
       dessert: "ice cream"}
    }

def return_course(diners, name, course)

return diners[:sarah][:main]

end
p return_course(diners, "sarah", "main")
```

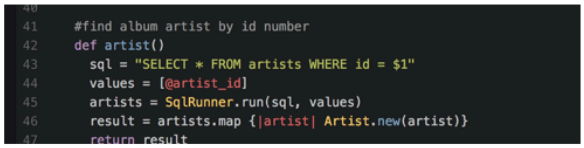


```
➜ week_2 git:(master) ✗ ruby hash_evidence.rb
"steak and chips"
➜ week_2 git:(master) ✗
```

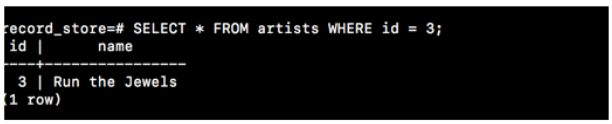
This screenshot shows the use of a hash, with a function accessing the keys of “Sarah” and “main” to print the value of this “Steak and Chips” to the terminal.

Week 3

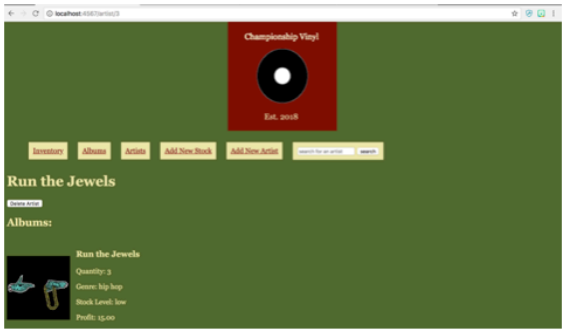
Unit	Ref	Evidence
I&T	I.T.3	Demonstrate searching data in a program. Take screenshots of: *Function that searches data *The result of the function running



```
#find album artist by id number
def artist()
  sql = "SELECT * FROM artists WHERE id = $1"
  values = [@artist_id]
  artists = SqlRunner.run(sql, values)
  result = artists.map {|artist| Artist.new(artist)}
  return result
```



```
record_store=# SELECT * FROM artists WHERE id = 3;
 id |       name
----+------------------
  3 | Run the Jewels
(1 row)
```

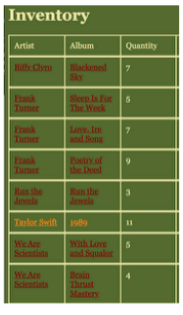


This screenshot shows a function that searches for all music artists in a database within a program by id number and the result of using the search in psql and a web app made using Sinatra.

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

```
#sort artists in alphabetical order
def self.sort_all()
  sql = "SELECT * FROM artists ORDER BY name;"
  artists = SqlRunner.run( sql )
  result = artists.map { |artist| Artist.new(artist) }
  return result
end
```

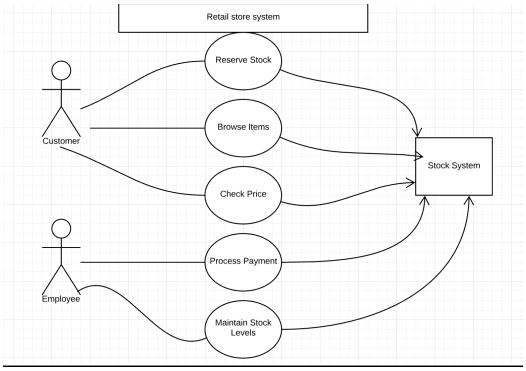
```
record_store=# SELECT * FROM Artists;
 id |       name
----+-------------------
  1 | Frank Turner
  2 | We Are Scientists
  3 | Run the Jewels
  4 | Taylor Swift
  5 | Biffy Clyro
(5 rows)

record_store=# SELECT * FROM artists ORDER BY name;
 id |       name
----+-------------------
  5 | Biffy Clyro
  1 | Frank Turner
  3 | Run the Jewels
  4 | Taylor Swift
  2 | We Are Scientists
(5 rows)
```



This screenshot shows a function that searches for all musical artists in a database and sorts them in alphabetical order by name, with the results in psql and a web app.

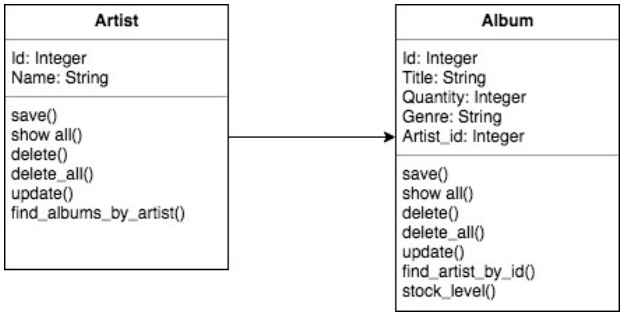
Week 5

Unit	Ref	Evidence
A&D	A.D.1	A Use Case Diagram



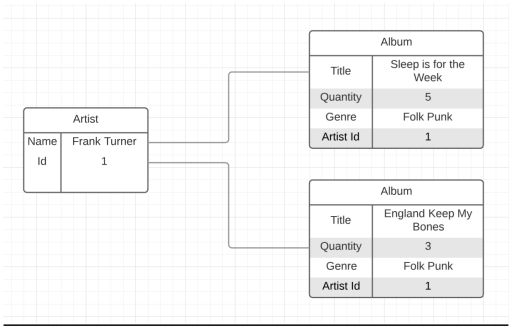
This is an example of a use case diagram for a retail store system that allows both staff and customers to interact with it.

Unit	Ref	Evidence
A&D	A.D.2	A Class Diagram



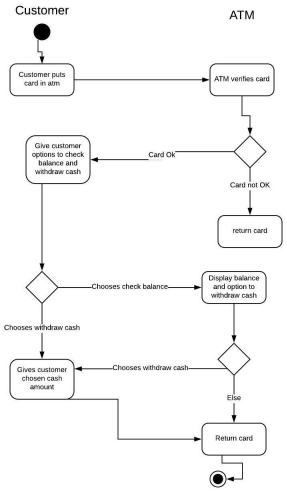
This is an example of a class diagram for a program which has two classes - one of Artists and one of Albums, with different properties and methods in each.

Unit	Ref	Evidence
A&D	A.D.3	An Object Diagram



This is an example of an object diagram in which one musical artist has many albums.

Unit	Ref	Evidence
A&D	A.D.4	An Activity Diagram



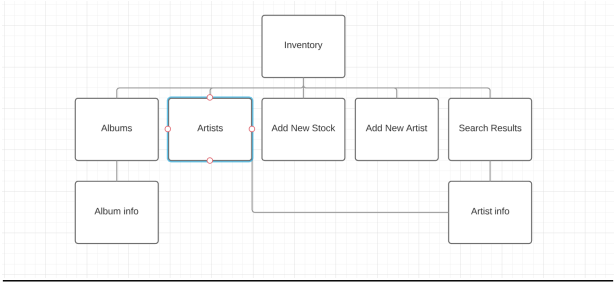
This is an activity diagram which shows the process of a customer using an ATM to check their balance and withdraw money.

Unit	Ref	Evidence
A&D	A.D.6	Produce an Implementations Constraints plan detailing the following factors: *Hardware and software platforms *Performance requirements *Persistent storage and transactions *Usability *Budgets *Time

Topic	Possible Effect of Constraint on Product	Solution
Hardware and software platforms	Product may not be compatible with all machines or browsers	Test on a variety of different machines and browsers to ensure compatibility.
Performance requirements	Product may require a certain type of processor or OS to run	Ensure product is compatible with all current OS and processors.
Persistent storage and transactions	Product may require a lot of memory or data to run.	Ensure all files are as small as they possibly can be.
Usability	Product may have usability issues for people with disabilities or may not be completely user friendly.	Use tools such as alt tags on images to make product as accessible as possible for all, carry out user tests to see where things need to change to make product more user friendly.
Budgets	Not enough money to complete product exactly how the client wants it to be completed.	More money, change the product slightly to fit the budget, find ways to save money and still deliver the best possible product
Time	Product may not reach its full potential in the given time frame as there is only time to implement the basics	More time

An implementations constraints plan for a program.

Unit	Ref	Evidence
P	P.5	User Site Map



A user sitemap for a record store inventory web app.

Unit	Ref	Evidence
P	P.6	2 Wireframe Diagrams

Wireframe diagrams showing an inventory page and an add new artist page for a record store web app.

Unit	Ref	Evidence
P	P.10	Example of Pseudocode used for a method

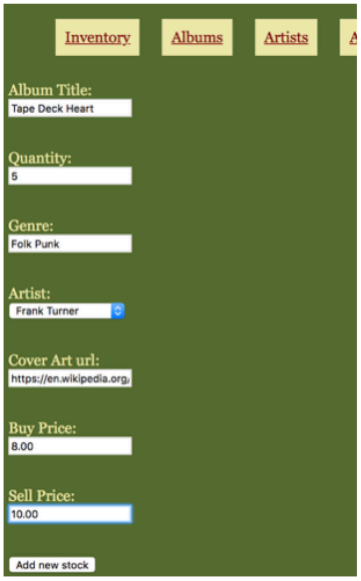
```
#method to determine whether stock level is low, medium or high, or out of stock

def stock_level()
  #if the quantity is greater than or equal to 10
  if @quantity >= 10
    #then return high
    return "high"
    #else if the quantity is between 5 and 9
  elsif @quantity >= 5 && @quantity <= 9
    #then return medium
    return "medium"
    #else if the quantity is less than 4 but greater than 0
  elsif @quantity <= 4 && @quantity > 0
    #then return low
    return "low"
    #else if the quantity is less than 0
  elsif @quantity <= 0
    #then return out of stock
    return "out of stock"
  end
end
```

Pseudocode for a function that returns low, medium, high or out of stock based on stock level.

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

A user inputting album information for a new album into a program, then that album appearing in the inventory.

```
<div class="new">

  <form action="/album" method="post">
    <label for="title">Album Title: </label><br>
    <input type="text" id="title" name="title"><br>

    <label for ="quantity"> Quantity: </label><br>
    <input type="number" id="quantity" name="quantity"><br>

    <label for="genre">Genre: </label><br>
    <input type="text" id="genre" name="genre"><br>

    <label for="artist"> Artist: </label><br>
    <select name="artist_id" id="artist_id">
      <% @artists.each do |artist| %>
      <option value="<%=artist.id%>"><%=artist.name%></option>
      <% end %>
    </select><br>

    <label for="url"> Cover Art url: </label><br>
    <input type="text" id="url" name="url"><br>

    <label for="buy_price"> Buy Price: </label><br>
    <input type="text" id="buy_price" name="buy_price"><br>

    <label for="sell_price"> Sell Price: </label><br>
    <input type="text" id="sell_price" name="sell_price"><br>

    <input type="submit" value="Add new stock">

  </form>
```



Frank Turner	Love, Ire and Song	7	folk punk	medium	£4.50	£10.00	£5.50	£38.50
Frank Turner	Poetry of the Deed	9	folk punk	medium	£4.50	£9.00	£4.50	£40.50
Frank Turner	Tape Deck Heart	5	Folk Punk	medium	£8.00	£10.00	£2.00	£10.00
Frank Turner	Sleep Is For The Week	2	folk punk	low	£5.50	£10.00	£4.50	£9.00
Run the Jewels	Run the Jewels	3	hip hop	low	£5.00	£10.00	£5.00	£15.00

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

```
#save albums
def save()
  sql = "INSERT INTO albums (title, quantity, genre, artist_id, url, buy_price, sell_price)
  VALUES ($1, $2, $3, $4, $5, $6, $7) RETURNING id;"
  values = [@title, @quantity, @genre, @artist_id, @url, @buy_price, @sell_price]
  album = SqlRunner.run(sql, values).first
  @id = album['id'].to_i
end
```

Seed data for a program with the save function and the result of that save function storing the data to the database in psql.

Unit	Ref	Evidence
P	P.15	Show the correct output of results and feedback to user. Take a screenshot of: * The user requesting information or an action to be performed * The user request being processed correctly and demonstrated in the program

Screenshots of the code used for a user to delete an album from a web app and the result of that album being deleted from the inventory.

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

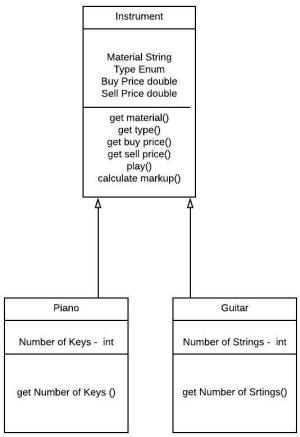
Examples of a test to get the total value when adding two cards together failing and then passing when the code has been corrected.

Week 7

Unit	Ref	Evidence
I&T	I.T.7	The use of Polymorphism in a program and what it is doing.

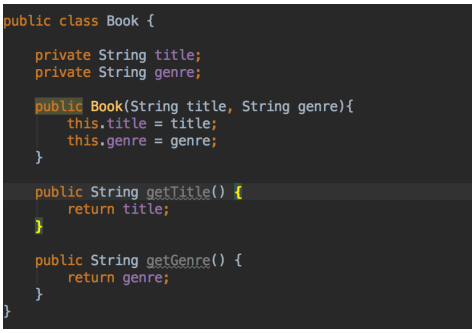
All instruments and accessories in the music shop implement the Sellable interface and therefore can be used in functions. All instruments and accessories can be added to the array list of items in the shop by using the object type Sellable in functions.

Unit	Ref	Evidence
A&D	A.D.5	An Inheritance Diagram



An Inheritance Diagram showing a Piano and a Guitar inheriting from an Instrument class.

Unit	Ref	Evidence
I&T	I.T.1	The use of Encapsulation in a program and what it is doing.



```
public class Book {

    private String title;
    private String genre;

    public Book(String title, String genre){
        this.title = title;
        this.genre = genre;
    }

    public String getTitle() {
        return title;
    }

    public String getGenre() {
        return genre;
    }
}
```

An example of a book class with private properties, preventing direct access to those properties outwit the class, with getter methods that can be used elsewhere in the program in order to access these values.

Unit	Ref	Evidence
I&T	I.T.2	Take a screenshot of the use of Inheritance in a program. Take screenshots of: *A Class *A Class that inherits from the previous class *An Object in the inherited class *A Method that uses the information inherited from another class.

```java
public abstract class Instrument implements Playable, Sellable {

    private String material;
    private String colour;
    private InstrumentType instrumentType;
    private String make;
    private double buyPrice;
    private double sellPrice;

    public Instrument(String material, String colour, InstrumentType instrumentType, String make, double buyPrice, double sellPrice) {
        this.material = material;
        this.colour = colour;
        this.instrumentType = instrumentType;
        this.make = make;
        this.buyPrice = buyPrice;
        this.sellPrice = sellPrice;
    }

    public String getMaterial() {
        return material;
    }

    public String getColour() {
        return colour;
    }
```

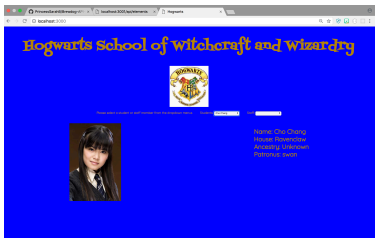
```java
public class Piano extends Instrument {

    private int numberOfKeys;

    public Piano(String material, String colour, InstrumentType instrumentType, String make, double buyPrice, double sellPrice, int numberOfKeys) {
        super(material, colour, instrumentType, make, buyPrice, sellPrice);
        this.numberOfKeys = numberOfKeys;
    }

    public int getNumberOfKeys() {
        return numberOfKeys;
    }

    public String play() {
        return "plinky plonky";
    }
}
```

```java
public class InstrumentTest {

    Instrument guitar;
    Instrument piano;

    @Before
    public void setUp() throws Exception {
        guitar = new Guitar("wood", "blue", InstrumentType.STRING, "Fender", 200.00, 250.00, 6);
        piano = new Piano("wood", "brown", InstrumentType.KEYBOARD, "Yamaha", 900.00, 1200.00, 88);
    }

    @Test
    public void hasMaterial() { assertEquals("wood", piano.getMaterial()); }

    @Test
    public void hasColour() { assertEquals("blue", guitar.getColour()); }

    @Test
    public void hasInstrumentType() { assertEquals(InstrumentType.STRING, guitar.getInstrumentType()); }

    @Test
    public void hasMake() { assertEquals("Yamaha", piano.getMake()); }
```

Screenshots of Instrument class and Piano class which inherits from Instrument, with examples of a new Piano object being used to test the get material and get make methods.

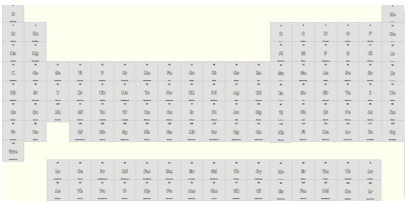
Week 10

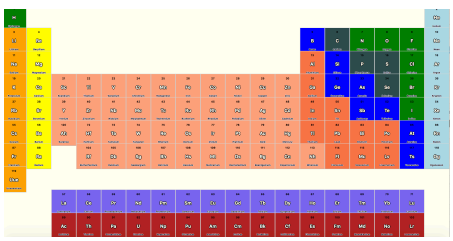
Unit	Ref	Evidence
P	P.11	Take a screenshot of one of your projects where you have worked alone and attach the Github link.

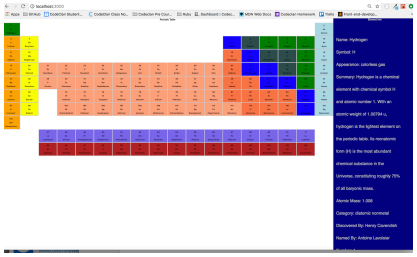


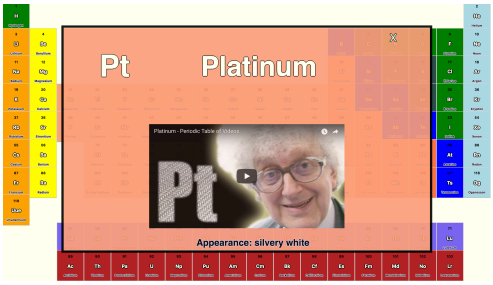
https://github.com/PrincessSarahB/Harry-Potter-API-Homework

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









Planning on white board and different stages of project right up to final version with pop up window when element is clicked.

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.

```
public ArrayList listVacantRooms() {
    ArrayList vacantRooms = new ArrayList<>();
    for (Room room : this.rooms) {
        if (room.guests.size() == 0) {
            vacantRooms.add(room);
        }
    }
    return vacantRooms;
}
```

I have chosen to use the above algorithm as it seemed like the best way to address the problem of finding out which rooms were empty in the hotel homework project.

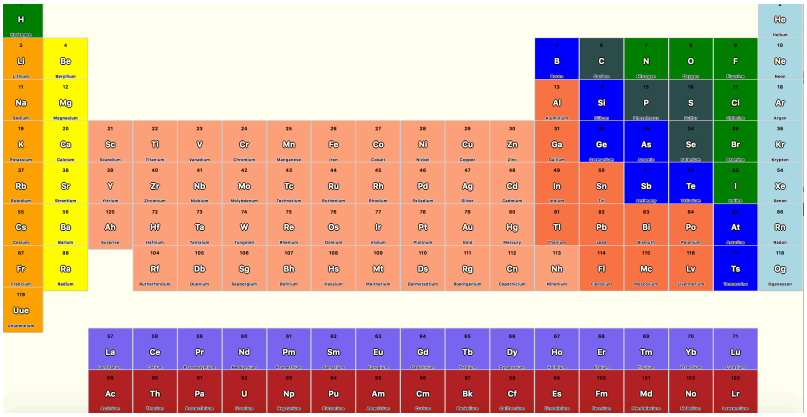
```
class SongList extends React.Component {
  render(){
    const songList = this.props.songs.map((song, index) => (
      <li key={song.id.attributes["im:id"]}><img src={song["im:image"][1].label} alt="song art" />
      {song.title.label}</li>
    ))

    return(
        <ol className='chart-list'>{songList}</ol>
      )
    }
  }


    export default SongList;
```

I have chosen to use this algorithm to map over an array of songs from an API and display each song's cover art and title in an ordered list to see what is in which position in the iTunes charts.

Week 12

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

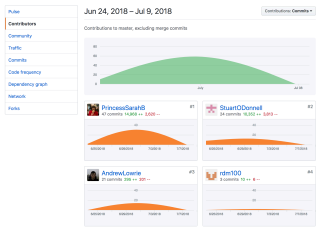
```
// Database api request gets a list of all elements in JSON format.
componentDidMount(){
  const url = '/api/elements';
  fetch(url).then(res => res.json()).then(elements => this.setState({elements: elements,
    elementToDisplay: elements[0]}));
    document.getElementsByClassName('info-box')[0].hidden = true;
}
```



Code that gets data from API and data being displayed in app.

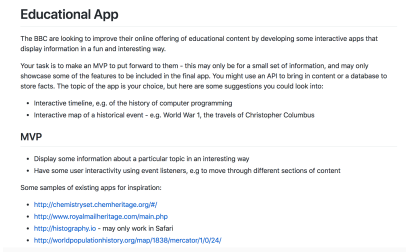
Week 15

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.



Contributors page showing I worked with Andrew, Roberto and Stuart.

Unit	Ref	Evidence
P	P.2	Take a screenshot of the project brief from your group project.



Educational App

The BBC are looking to improve their online offering of educational content by developing some interactive apps that display information in a fun and interesting way.

Your task is to make an MVP 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. You might use an API to bring in content or a database to store facts. The topic of the app is your choice, but here are some suggestions you could look into:

- Interactive timeline, e.g. of the history of computer programming
- Interactive map of a historical event - e.g. World War 1, the travels of Christopher Columbus

MVP

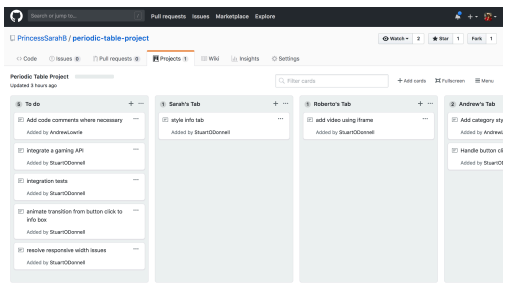
- Display some information about a particular topic in an interesting way
- Have some user interactivity using event listeners, e.g to move through different sections of content

Some samples of existing apps for inspiration:

- http://chemistryset.chemheritage.org/#/
- http://www.royalmailheritage.com/main.php
- http://histography.io - may only work in Safari
- http://worldpopulationhistory.org/map/1838/mercator/1/0/24/

Project brief from group project.

Unit	Ref	Evidence
P	P.3	Provide a screenshot of the planning you completed during your group project, e.g. Trello MOSCOW board.



Project board from Periodic Table of Elements group project.

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

Acceptance Criteria Table

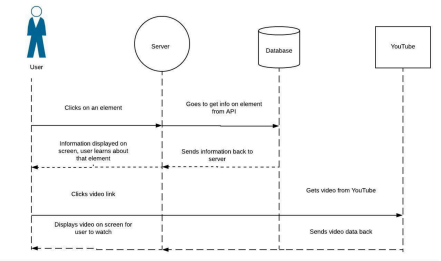
Acceptance Criteria	Expected Result/ Output	Pass/Fail
A user can click an element to get information about it	Information about the element is displayed in the designated area on the page	Fail
A user can click a link on the element information	Redirects to wikipedia page	Fail
When a user hovers over an element it is animated	Element pops out or changes colour upon mouse hover	Fail
Element of the day feature on screen	Every day a different element is displayed in a designated area on screen, element changes each day	Fail
A user can play an interactive game to guess the element	Element symbol to guess appears and a correct message displays when correct one is picked	Fail

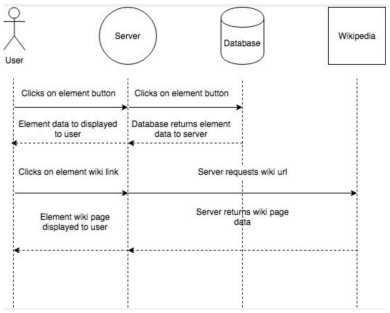
Acceptance Criteria Table

Acceptance Criteria	Expected Result/ Output	Pass/Fail
A user can click an element to get information about it	Information about the element is displayed in the designated area on the page	Pass
A user can click a link on the element information	Redirects to wikipedia page	Pass
When a user hovers over an element it is animated	Element pops out or changes colour upon mouse hover	Pass
Element of the day feature on screen	Every day a different element is displayed in a designated area on screen, element changes each day	Fail
A user can play an interactive game to guess the element	Element symbol to guess appears and a correct message displays when correct one is picked	Fail

Failing, then passing acceptance criteria plan

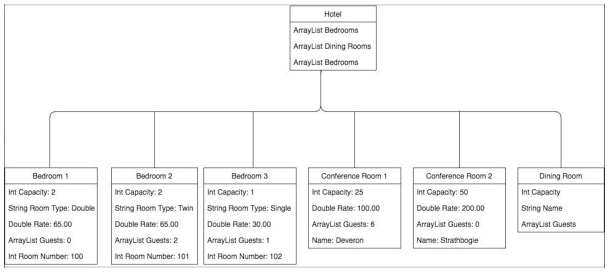
Unit	Ref	Evidence
P	P.7	Produce two system interaction diagrams (sequence and/or collaboration diagrams).



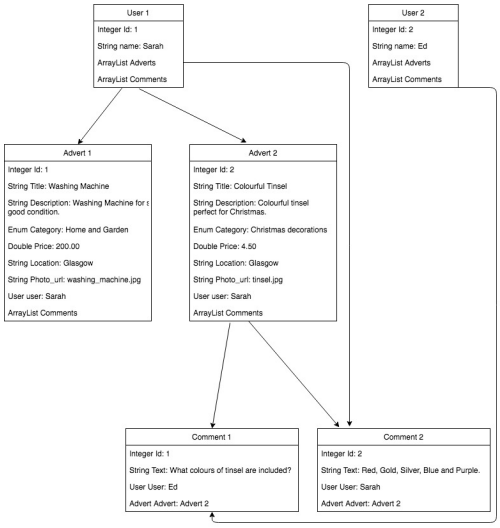


Two sequence diagrams

Unit	Ref	Evidence
P	P.8	Produce two object diagrams.



An object diagram showing a hotel with many rooms of different types.



An Object diagram showing users, adverts and comments. One user has many adverts and comments and one advert has many comments by different users.

Unit	Ref	Evidence
P	P.17	Produce a bug tracking report

Were unable to retrieve information from seeds	Fail	We were passing in wrong variable in props	Pass
User could not select element info by clicking element	Fail	Added onClick function to each element button	Pass
Wrong element information showing on button click	Fail	Changed onevent target value to event.currentTarget.value	Pass
Iframe not displaying youtube link	Fail	Wrote a function to split id from youtube link and append it to youtube embed code	Pass
Unable to add colours to element groups	Fail	Added group class to each element from api for css to select for styling	Pass
Video keeps playing after popup window is closed	Fail	Setting iframe src to "" in handleCloseButton	Pass

A bug tracking report for Periodic Table of Elements project.