

Kathryn Kiernan - Implementation and Testing Unit (SQA PDA: Software Development)

I.T a - Encapsulation

```
class Hero
attr_reader :name, :title, :weapon

def initialize(name, title, weapon)
  @name = name
  @title = title
  @weapon = weapon
end

def attack
  return "#{title} #{name} weilds his mighty #{weapon}"
end
```

I.T b - Use of Inheritance

```
var Villain = function(villainName, age, health, weapon, minions){
  this.villainName = villainName;
  this.age = age;
  this.health = health;
  this.weapon = weapon;
  this.minions = [];
}

Villain.prototype = {
  addMinion: function(minion){
    return this.minions.push(minion);
  },
  createArmy: function(minion){
    var x = 0;
    while(x < 100) {
      this.minions.push(minion);
      x = x + 1;
    }
  },
  curse: function(hero){
    hero.health = hero.health - (this.age * 0.002);
  },
  attack: function(hero){
    var attackDamage = this.weapon.damage + (this.age * 0.002);
    return hero.healthDamage = attackDamage;
  }
}

module.exports = Villain;
```

```
it('should create an army of rats', function(){
  var rattigan = new Rat("Rattigan", "brown");
  var saruman = new Villain("Saruman", 2034, 25, "staff");
  saruman.createArmy(rattigan);
  assert.equal(100, saruman.minions.length);
});

it('should damage the hero with a curse', function(){
  var staff = new Weapon("Staff", 10);
  var saruman = new Villain("Saruman", 2034, 25, staff);
  var legolas = new Hero("Legolas", 20, "Lembas bread");
  saruman.curse(legolas);
  assert.closeTo(15.93, legolas.health, 0.01)
});

it('should damage the hero with an attack', function(){
  var club = new Weapon("Club", 1);
  var orc = new Villain("Orc", 4, 15, club);
  var legolas = new Hero("Legolas", 4, "Lembas bread");
  orc.attack(legolas);
  legolas.canBlock(orc);
  assert.equal(2.992, legolas.health)
});
```

I.T c (i) - Search function and results

```
def index
  @movies = Movie.where(user_id: current_user.id)
  if !params[:search].blank?
    @omdb_movies = Movie.search(params[:search])
  else
    @omdb_movies = Movie.search(params[:search]) if params[:search]
  end
end
```

My Movies

Type a word or name related to a film into the search below. It will list all films found on IMDB that contains that word or name. Once you have found a movie that you like, you can save it to your Must See. Simply click the 'Add Movie' button and you're done!

phantom of the opera

Search

Home > My Movies

All Movies

phantom of the opera

Movies Found

Movie	Year	Actions
The Phantom of the Opera	2004	Movie Details Add Movie
The Phantom of the Opera	1925	Movie Details Add Movie
Phantom of the Opera	1943	Movie Details

I.T c (ii) Sort function and results

```

1 class Wine < ActiveRecord::Base
2
3   mount_uploader :image, WineImageUploader
4
5   belongs_to :user
6   belongs_to :grape
7   belongs_to :colour
8   belongs_to :region
9   belongs_to :cellar
10  has_many :lists, through: :listings
11
12  validates_presence_of :make, :colour, :grape
13
14  def select_wine_details
15    "#{self.make}" + " : " + "#{grape.grape_name}" + " " + "#{self.vintage_year}"
16  end
17
18  def human_readable_purchase_date
19    date_purchased.strftime('%d %B %Y') if date_purchased
20  end
21
22  def self.order_list(sort_order)
23    if sort_order == "newest" || sort_order.blank?
24      order(created_at: :desc)
25    elsif sort_order == "make"
26      order(make: :asc)
27    elsif sort_order == "rating"
28      @wines = Wine.all.sort_by { |w| -(w.rating.to_i) }
29    else
30      order(created_at: :asc)
31    end
32  end
33
34  end

```

```

<div class="container">
  <div class="row" id = "page-link-trail">
    <div class="box-7">
      <p>&lt;= link_to 'Home', root_path %> Journal</p>
    </div>
    <div class="box-3 link-button">
      <p>&lt;= link_to 'Add a New Wine', new_wine_path %></p>
    </div>
  </div>
  <div class="box-2 dropdown" id="align-right">
    <div class="dropbtn">Sort By ></div>
    <div class="dropdown-content">
      <a href="<%= wines_path(sort_by: 'oldest') %>">Oldest</a>
      <a href="<%= wines_path(sort_by: 'newest') %>">Newest</a>
      <a href="<%= wines_path(sort_by: 'make') %>">Make</a>
      <a href="<%= wines_path(sort_by: 'rating') %>">Rating</a>
    </div>
  </div>
</div>

```

Home > Journal

Add a New Wine

Sort By >

- Oldest
- Newest
- Make
- Rating

Casillero Del Diablo

Grape: Sauvignon Blanc
Vintage: 2015
Region: Chile, Americas
Class: Still
Price: £5
Sainsbury's

Blossom Hill

Grape: Sauvignon Blanc
Vintage: 2015
Region: USA, Americas
Class: Still
Price: £5
Sainsbury's

First Cape

Grape: Pinot Noir
Vintage: 2015
Region: Bolivia, Americas
Class: Still
Price: £4
Tesco

Black Tower

Grape: Zweigelt
Vintage: 2015
Region: Venezuela, Americas
Class: Still
Price: £7
Sainsbury's

Casillero Del Diablo

Grape: Sauvignon Blanc
Vintage: 2015
Region: Chile, Americas
Class: Still
Price: £5
Sainsbury's

Echo Falls

Grape: Pinot Grigio
Vintage: 2015
Region: Spain, Europe
Class: Still
Price: £3
Tesco

Isla Negra

Grape: Cabernet Sauvignon
Vintage: 2015
Region: Chile, Americas
Class: Still
Price: £7
Sainsbury's

Black Tower

Grape: Zweigelt
Vintage: 2015
Region: Venezuela, Americas
Class: Still
Price: £5
Sainsbury's

Blossom Hill

Grape: Sauvignon Blanc
Vintage: 2015
Region: USA, Americas
Class: Still
Price: £5
Sainsbury's

Casillero Del Diablo

Grape: Sauvignon Blanc
Vintage: 2015
Region: Chile, Americas
Class: Still
Price: £5
Sainsbury's

Gallo

Grape: Vignoles
Vintage: 2015
Region: Cape Verde, Africa
Class: Still
Price: £4
Tesco

Hardy's

Grape: Syrah
Vintage: 2015
Region: Portugal, Europe
Class: Still
Price: £5
Sainsbury's

Echo Falls

Grape: Pinot Grigio
Vintage: 2015
Region: Spain, Europe
Class: Still
Price: £3
Tesco

First Cape

Grape: Pinot Noir
Vintage: 2015
Region: Venezuela, Americas
Class: Still
Price: £4
Tesco

Isla Negra

Grape: Cabernet Sauvignon
Vintage: 2015
Region: Chile, Americas
Class: Still
Price: £7
Sainsbury's

I.T d (i) Array

```

3
4  var Game = function(computerTally, gameWinner, playerArray, computerCards, computerTurnCard){
5    this.computerTally = computerTally;
6    this.gameWinner = gameWinner;
7    this.playerArray = [];
8    this.computerCards = [];
9    this.computerTurnCard = null;
10   this.characters = characterData;
11 }
12

```

```

setupCards: function () {
  var gameCards = _.shuffle(characterData);
  while(this.computerCards.length < 7){
    this.addCard(gameCards.pop());
  }
  this.playerArray.forEach(function (player){
    var x = 0;
    while(x < 6) {
      player.giveCard(gameCards.pop());
      x = x + 1;
    }
  })
},

```

```

→ top_trumps mocha specs

game
✓ should create a game with an empty player array
✓ should create a game with an empty array of computer cards
✓ should create a new game with an array of one player
✓ should have an array of characters
✓ should shuffle the characters and add 6 characters to the players character array

```

I.T d (ii) Hash

```

characterData.js
1 var characterData = [{"name": "Aragorn",
2   "realm": "Middle Earth",
3   "culture": "Hero",
4   "age": 87,
5   "power": 13,
6   "wisdom": 7,
7   "courage": 9
8 },
9   {"name": "Gandalf",
10   "realm": "Middle Earth",
11   "culture": "Wizard",
12   "age": 2018,
13   "power": 20,
14   "wisdom": 18,
15   "courage": 17
16 },
17   {"name": "Magneto",
18   "realm": "The Marvel Universe",
19   "culture": "Mutant",
20   "age": 67,
21   "power": 18,
22   "wisdom": 17,
23   "courage": 15
24 },

```

```

compareAge: function(player){
  if(this.computerTurnCard[0].age > player.playerTurnCard[0].age){
    this.computerTally += 1;
  } else {
    player.playerTally += 1;
  }
  player.resetPlayerTurnCard();
  return this.computerTurnCard = null;
},

```

```

top_trumps mocha specs

game
✓ should create a game with an empty player array
✓ should create a game with an empty array of computer cards
✓ should create a new game with an array of one player
✓ should have an array of characters
✓ should shuffle the characters and add 6 characters to the players character array
✓ should add 6 cards to the computer cards array
✓ should take a card from the computers cards and push it into the computers turn card array
✓ should take a card from the players cards and push it into the players turn card array
✓ should should set the computer card back to null - age
✓ should should set the player card back to null - age
✓ should should set the player card back to null - power
✓ should should set the computer card back to null - power
✓ should should set the player card back to null - wisdom
✓ should should set the computer card back to null - wisdom
✓ should should set the player card back to null - courage
✓ should should set the computer card back to null - courage
✓ should decide that the computer is the winner
✓ should decide that the player is the winner
The Player is the Winner!
✓ should decide a winner - player
The Computer is the Winner!
✓ should decide a winner - computer

```