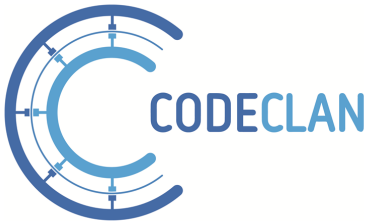


Evidence Gathering Document for SQA Level 8 Professional Developer Award.

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
		Description:

Screenshot:

```ruby
array_of_numbers = [1,2,3,4,5,6,7,8,9,16,36,15]

def calculate_total(numbers)
  total = 0
  for number in numbers
    total = total + number
  end
  return total
end

p "This is the sum of the array elements: #{calculate_total(array_of_numbers)}"
```

```
➜ lesson ruby quiz.rb
"This is the sum of the array elements: 112"
```

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
		Description:

Screenshot:

```ruby
margaret = {name: "Margaret", age: 2, eggs: 3}
hetty = {name: "Hetty", age: 1, eggs: 2}
henrietta = {name: "Henrietta", age: 3, eggs: 1}
audrey = {name: "Audrey", age: 2, eggs: 4}
mabel = {name: "Mabel", age: 5, eggs: 1}

chickens = [margaret,hetty,henrietta,audrey,mabel]

def find_animal_by_name(animals,name)
  found = false
  for animal in animals
    if animal[:name]==name
      found = true
    end
  end
  return "The animal was found: #{found}."
end

p find_animal_by_name(chickens,"James")
```

```
➜ lesson ruby loops_in_functions.rb
"The animal was found: false."
```

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
		Description:

Screenshot:

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
		Description:

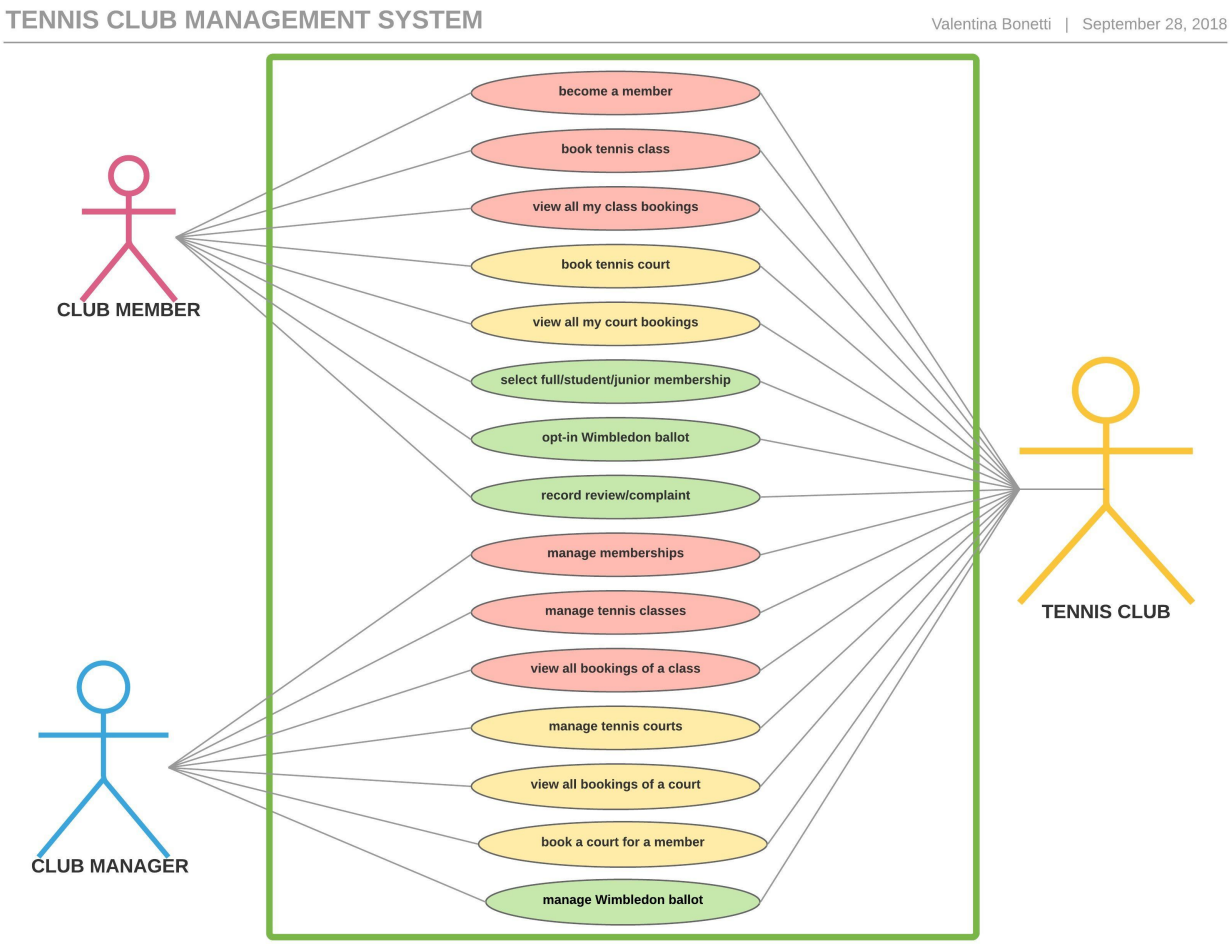
Paste Screenshot here

Description here

Week 5 and 6

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

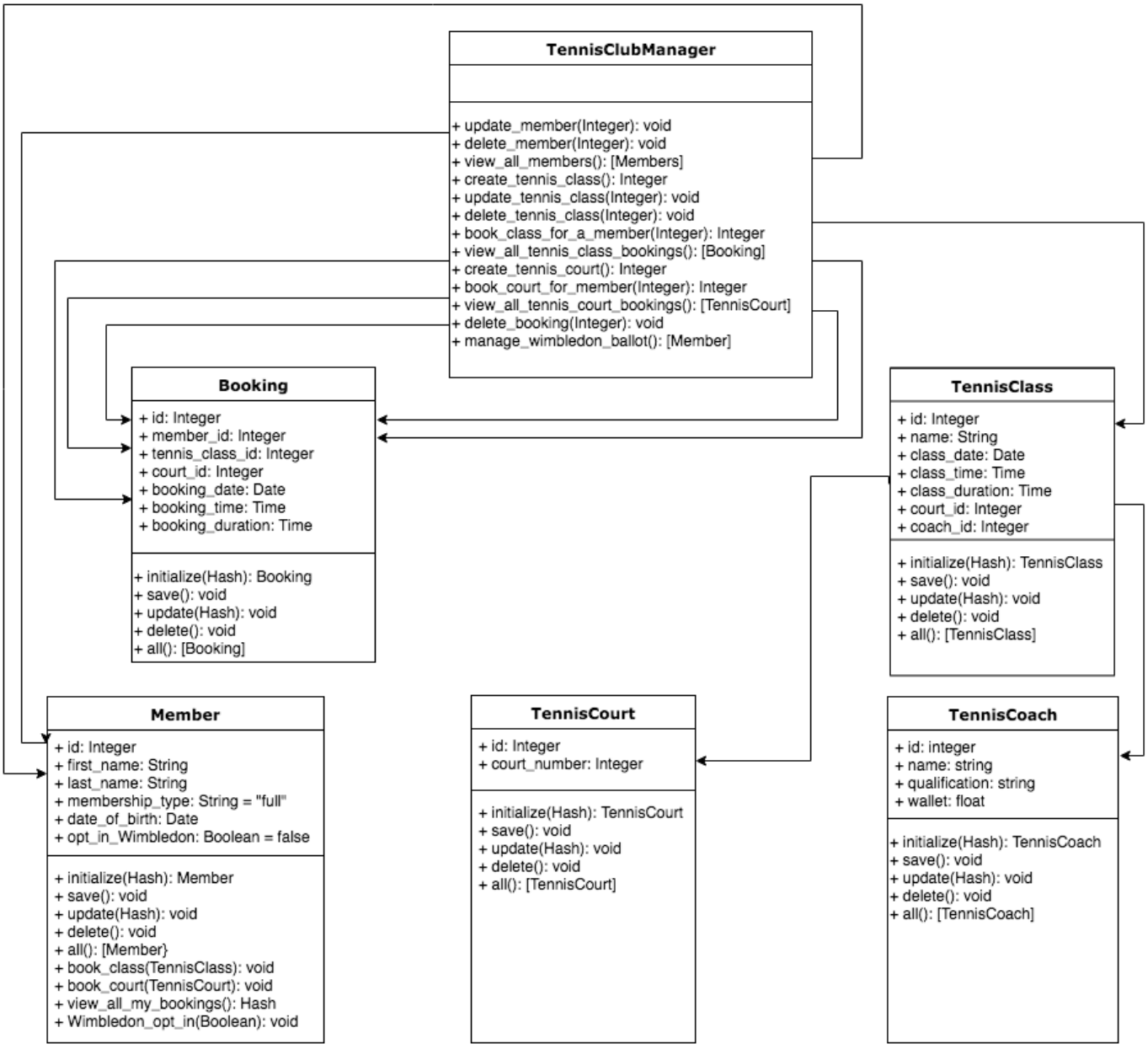
Screenshot:



WEEK 5 PROJECT: TENNIS CLUB MANAGEMENT SYSTEM

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

Screenshot:



WEEK 5 PROJECT: TENNIS CLUB MANAGEMENT SYSTEM, CLASS DIAGRAM.

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

Screenshot:

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

Screenshot:

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
		Description:

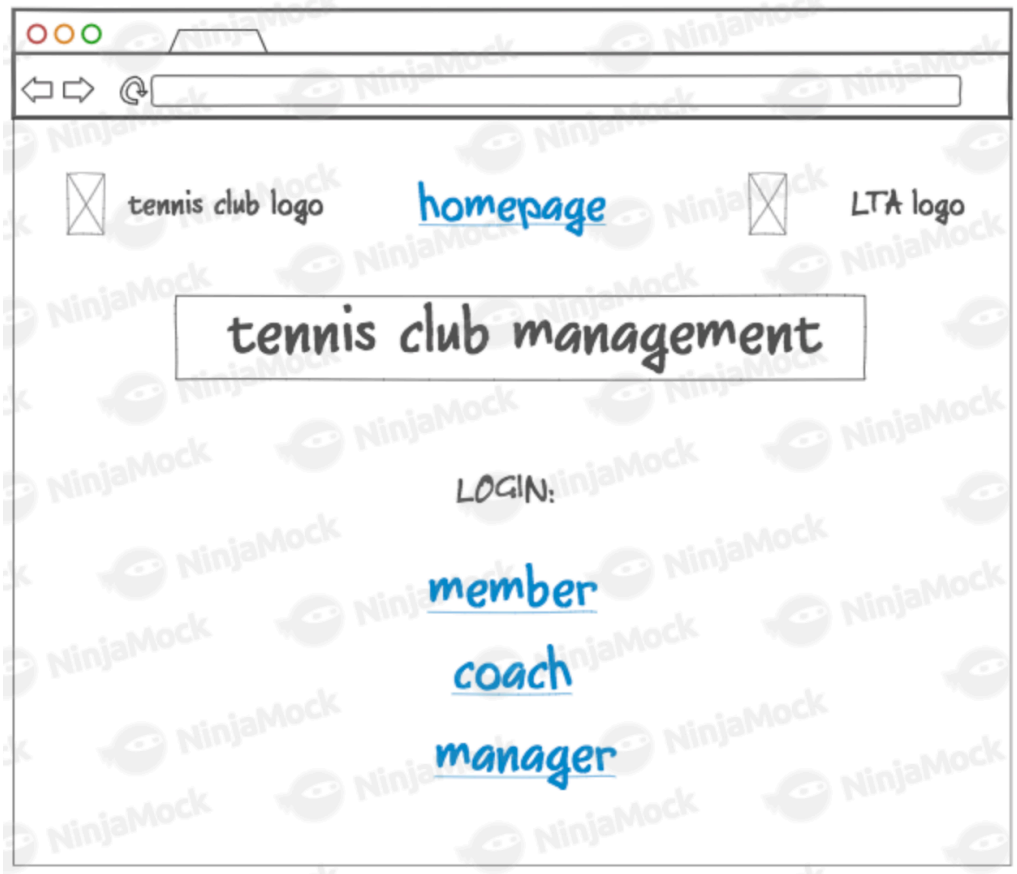
Screenshot:

Unit	Ref	Evidence
P	P.5	User Site Map
		Description:

Screenshot:

Unit	Ref	Evidence
P	P.6	2 Wireframe Diagrams
		Description:

Screenshot:



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

Paste Screenshot here

Description here

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
		Description:

Paste Screenshot here

Description here

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
		Description:

Paste Screenshot here

Description here

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
		Description:

Paste Screenshot here

Description here

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:

Paste Screenshot here

Description here

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

Description here

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
		Description:

Paste Screenshot here

Description here

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

```ruby

require_relative('card.rb')
class CardGame


  def checkforAce(card)
    if card.value = 1 # this needs to be == , otherwise always true
      return true
    else
      return false
    end
  end

  dif highest_card(card1 card2)
  #"def" instead of "dif", comma missing between card1 and card2
  if card1.value > card2.value
    return card.name
    # card object not defined
    # card class method name not defined
  else
    # what if they are equal? equal condition missing
    card2
    # return keyword missing. It would work anyway, but better with
    return
  end
  # bad indentation of if statement
end
# bad indentation of method end
end
# this last end closes the class... it should be deleted

def self.cards_total(cards)
  total
  # total should be set to an initial value (0)
  for card in cards
    total += card.value
    return "You have a total of" + total
    # this statement returns the value of the first card and
    # ends the for loop after the first iteration.
    # It needs to be placed outside the for loop, after the following
    end.
  end
end

```

P18.A EXAMPLE OF TEST CODE

```ruby
# tests for CardGame

def test_checkforAce_true()
  expected = true
  actual = @cardgame.checkforAce(@card3)
  assert_equal(expected,actual)
end

def test_checkforAce_false()
  expected = false
  actual = @cardgame.checkforAce(@card1)
  assert_equal(expected,actual)
end

def test_highest_card()
  expected = @card1
  actual = @cardgame.highest_card(@card1,@card3)
  assert_equal(expected,actual)
end

# This ranking is used in the game of bridge:
# spades (highest), hearts, diamonds, clubs (lowest)
def test_highest_card_samevalue()
  expected = @card1
  actual = @cardgame.highest_card(@card1,@card2)
  assert_equal(expected,actual)
end

def test_highest_card_valueMoreImportantThanSuit()
  expected = @card2
  actual = @cardgame.highest_card(@card2,@card3)
  assert_equal(expected,actual)
end

def test_cards_total()
  expected = "You have a total of 13"
  actual = CardGame.cards_total(@cards)
  assert_equal(expected,actual)
end
```

P18.A - TESTS

```
Run options: --seed 64526

# Running:

...F....

Finished in 0.001297s, 6168.0797 runs/s, 6168.0797 assertions/s.

  1) Failure:
TestCardGame#test_cards_total [specs/testing_task_2_spec.rb:68]:
Expected: "You have a total of 13"
  Actual: "You have a total of 2"

8 runs, 8 assertions, 1 failures, 0 errors, 0 skips
```

P18.B - EXAMPLE OF TEST FAILING

```ruby
require_relative('card.rb')
class CardGame


  def checkforAce(card)
    if card.value == 1
      return true
    else
      return false
    end
  end

  def highest_card(card1,card2)
    # bridge suit ranking low to high is alphabetical order:
    # low_to_high_suit_rank = ["clubs","diamonds","hearts","spades"]
    cards = [card1,card2]
    cards.sort_by! {|card| [card.value, card.suit]}
    cards.reverse!
    return cards[0]
  end


  def self.cards_total(cards)
    total = 0
    for card in cards
      total += card.value
    end
    return "You have a total of " + total.to_s
  end

end

```

P18.C - CORRECTED CODE

```
ruby specs/testing_task_2_spec.rb
Run options: --seed 53349

# Running:

........

Finished in 0.001078s, 7421.1507 runs/s, 7421.1507 assertions/s.

8 runs, 8 assertions, 0 failures, 0 errors, 0 skips
```

P18.D - ALL TEST PASSED

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.
		Description:

Paste Screenshot here

Description here

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

Paste Screenshot here

Description here

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

Paste Screenshot here

Description here

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

Paste Screenshot here

Description here

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

Paste Screenshot here

Description here

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

Paste Screenshot here

Description here

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

Paste Screenshot here

Description here

Week 12

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

Paste Screenshot here

Description here

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

Paste Screenshot here

Description here

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

Paste Screenshot here

Description here

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.
		Description:

Paste Screenshot here

Description here

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.
		Description:

Paste Screenshot here

Description here