

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: | |

```

→ ernest_fraiman_pda git:(master) ruby arrays.rb
["Yrucrem", "Sunev", "Htrae", "Sram", "Retipuj", "Nrutas", "Sunaru", "Enutpen"]
→ ernest_fraiman_pda git:(master) █

] card_game_spec.rb card_game.rb arrays.rb
1 # Demonstrate the use of an array in a program. Take screenshots of:
2 # *An array in a program
3 # *A function that uses the array
4 # *The result of the function running
5
6
7 planets = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus",
8 planets.push("Neptune")
9
10 def backwards_names(array)
11   for planet in array
12     planet.reverse!
13     planet.capitalize!
14   end
15   return array
16 end
17
18 p backwards_names(planets)
>

```

Array named planets is used in a function backwards_names to reverse each string in an array and capitalise reversed string.

| 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: | |

```

→ ernest_fraiman_pda git:(master) ✘ ruby week2_arrays_and_hashes_code/hashes.rb
7423529
→ ernest_fraiman_pda git:(master) ✘ ruby week2_arrays_and_hashes_code/hashes.rb
"The size of all the planets is 7423529"
→ ernest_fraiman_pda git:(master) ✘

arrays.rb — week2_arrays_code | hashes.rb | arrays.rb — week2_arrays_and_h...
1  # Demonstrate the use of a hash in a program. Take screenshots of:
2  # *A hash in a program
3  # *A function that uses the hash
4  # *The result of the function running
5
6  planets = {"Mercury" => 748,
7  "Venus" => 4602, "Earth" => 5101, "Mars" => 1448, "Jupiter" => 6142000,
8  "Saturn" => 427000, "Uranus" => 80830}
9
10 planets["Neptune"] = 761800
11
12 def sum_planet_sizes(hash)
13   sum = 0
14   hash.each do |planet, size|
15     sum += size
16   end
17   return "The size of all the planets is #{sum}"
18 end
19
20 p sum_planet_sizes(planets)
21

```

Hash of named planets containing string and integer (for name and size), function sum_planet_sizes takes in hash and returns sum of all the planets in hash size.

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: | |

```

◆ ernest_fraiman_pda git:(master) ✘ ruby week3_data_sorting_and
 _searching/data_search.rb
 {"name"=>"Venus", "size"=>4602}
◆ ernest_fraiman_pda git:(master) ✘

arrays.rb           hashes.rb          data_sort.rb
1 # Demonstrate searching data in a program. Take screenshots of:
2 # *Function that searches data
3 # *The result of the function running
4
5 planets = [{"name" => "Mercury", "size" => 748},
6 {"name" =>"Venus", "size" => 4602}, {"name" =>"Earth", "size" => 5101},
7 [{"name" =>"Mars", "size"=> 1448}, {"name" => "Jupiter", "size" => 6142000}
8 {"name" => "Saturn", "size"=> 427000}, {"name" => "Uranus", "size"=> 80830
9
10
11 def find_planet_by_name(array, planet_name)
12   match = "not found"
13   for planet in array
14     match = planet if(planet['name'] == planet_name)
15   end
16   return match
17 end
18
19 p find_planet_by_name(planets, "Venus")
20

```

Description: function called `find_planet_by_name`, searches data hash for the object with the same name and then return object if it exists.

| 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: |

```

◆ ernest_fraiman_pda git:(master) ✘ ruby week3_data_sorting_and_searching/data_sort.rb
Jupiter
Saturn
Uranus
Neptune
Earth
Venus
Mars
Mercury
◆ ernest_fraiman_pda git:(master) ✘

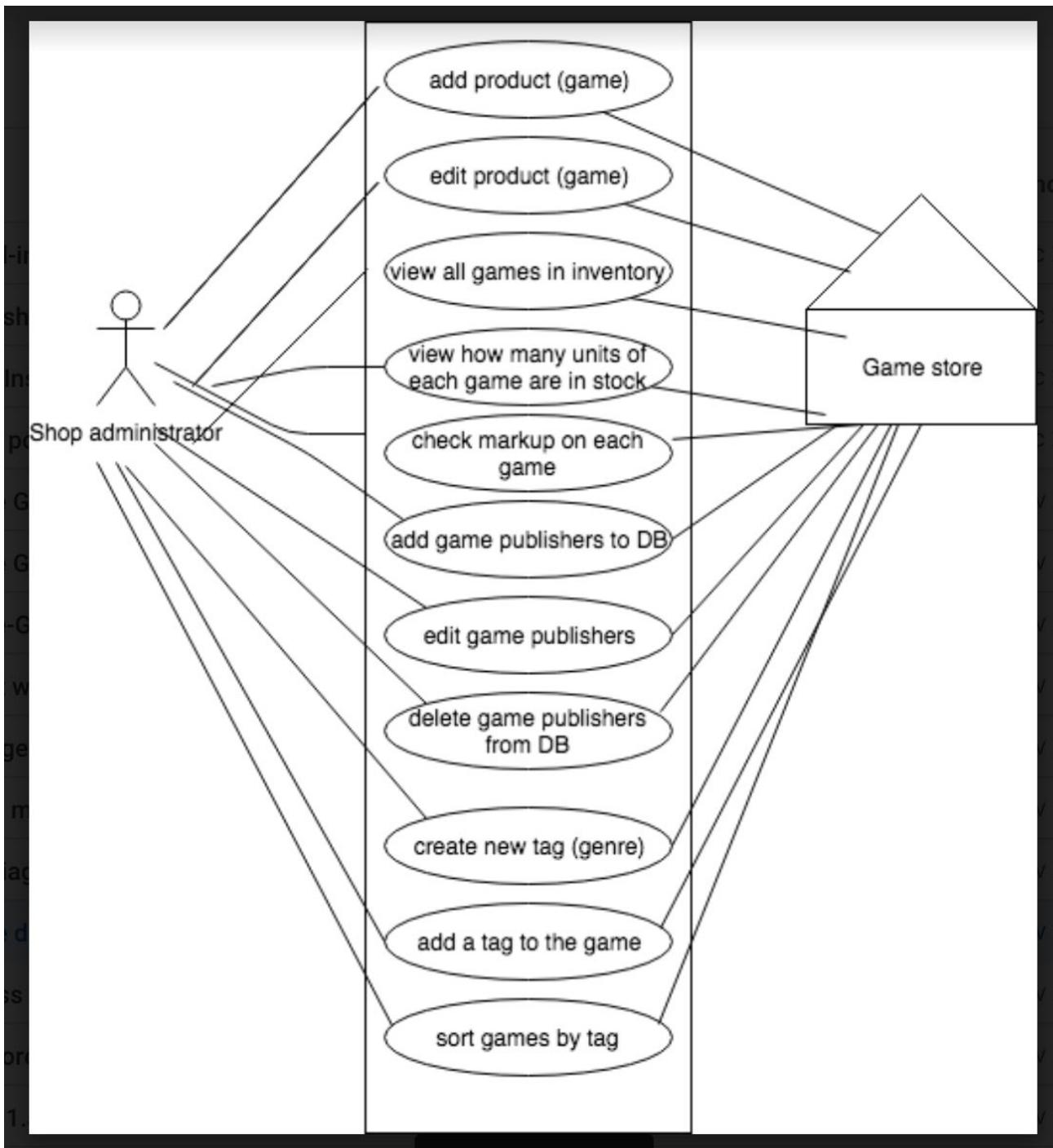
arrays.rb           hashes.rb          data_sort.rb
1 # Demonstrate sorting data in a program. Take screenshots of:
2 # *Function that sorts data
3 # *The result of the function running
4
5 planets = {"Mercury" => 748,
6 "Venus" => 4602, "Earth" => 5101, "Mars" => 1448, "Jupiter" => 6142000
7 "Saturn" => 427000, "Uranus" => 80830
8
9 planets["Neptune"] = 76180
10
11 def sum_planet_sizes(hash)
12   sum = 0
13   hash.each do |planet, size|
14     sum += size
15   end
16   return "The size of all the planets is #{sum}"
17 end
18
19 > def sort_planets_by_size(hash)
20   new_hash = hash.sort_by{|planet, size| size}
21   new_hash.reverse!
22   new_hash.each do |planet, size|
23     puts planet
24   end
25 end
26
27 sort_planets_by_size(planets)
28

```

Function `sort_planets_by_size`, sorts hash of planets by their size, print each planets name by the biggest first

Week 5 and 6

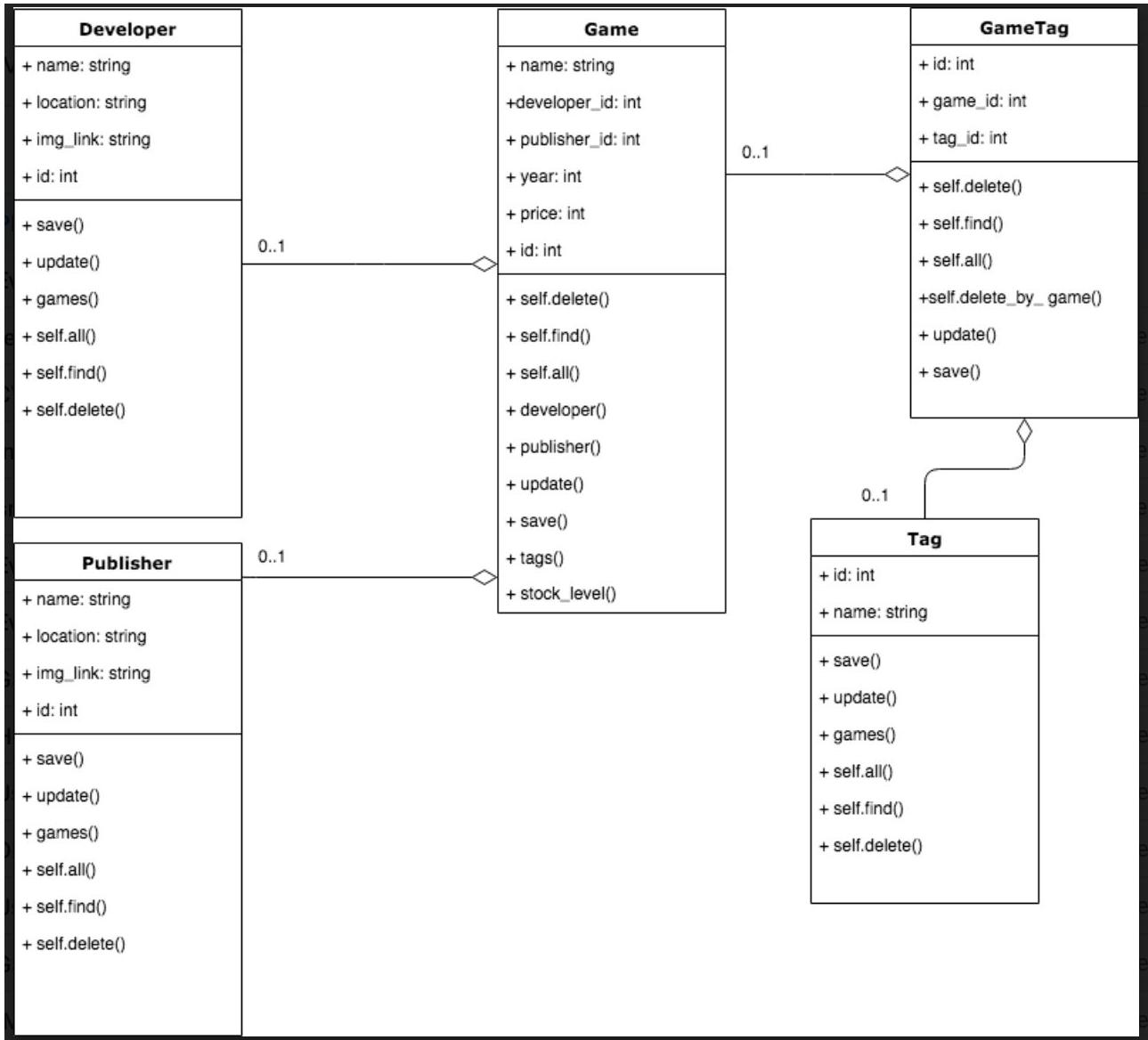
| Unit | Ref | Evidence | |
|------|-------|---------------------|--|
| A&D | A.D.1 | A Use Case Diagram | |
| | | Description: | |



Description:

Use case diagram for games store inventory management app, showing how shop administrator interacts with the system. User can add new game to the stock and/or edit, delete it. Create, edit, delete developers or publishers or genres to the database. Sort or view the whole shop inventory.

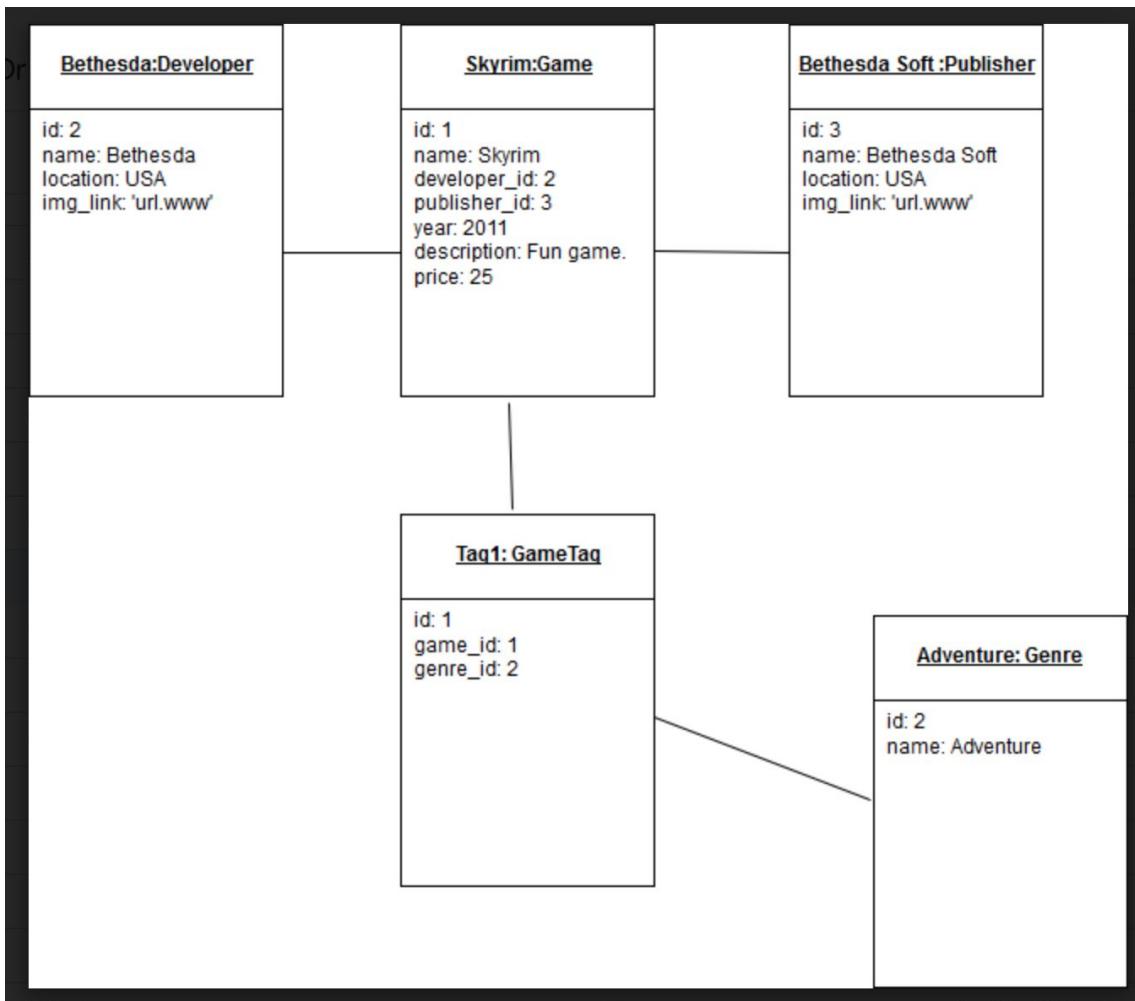
| Unit | Ref | Evidence | |
|------|-------|---------------------|--|
| A&D | A.D.2 | A Class Diagram | |
| | | Description: | |



Description:

Class diagram of game-store app. It shows the relationship between the classes, the class variables and the methods which are available to each class.

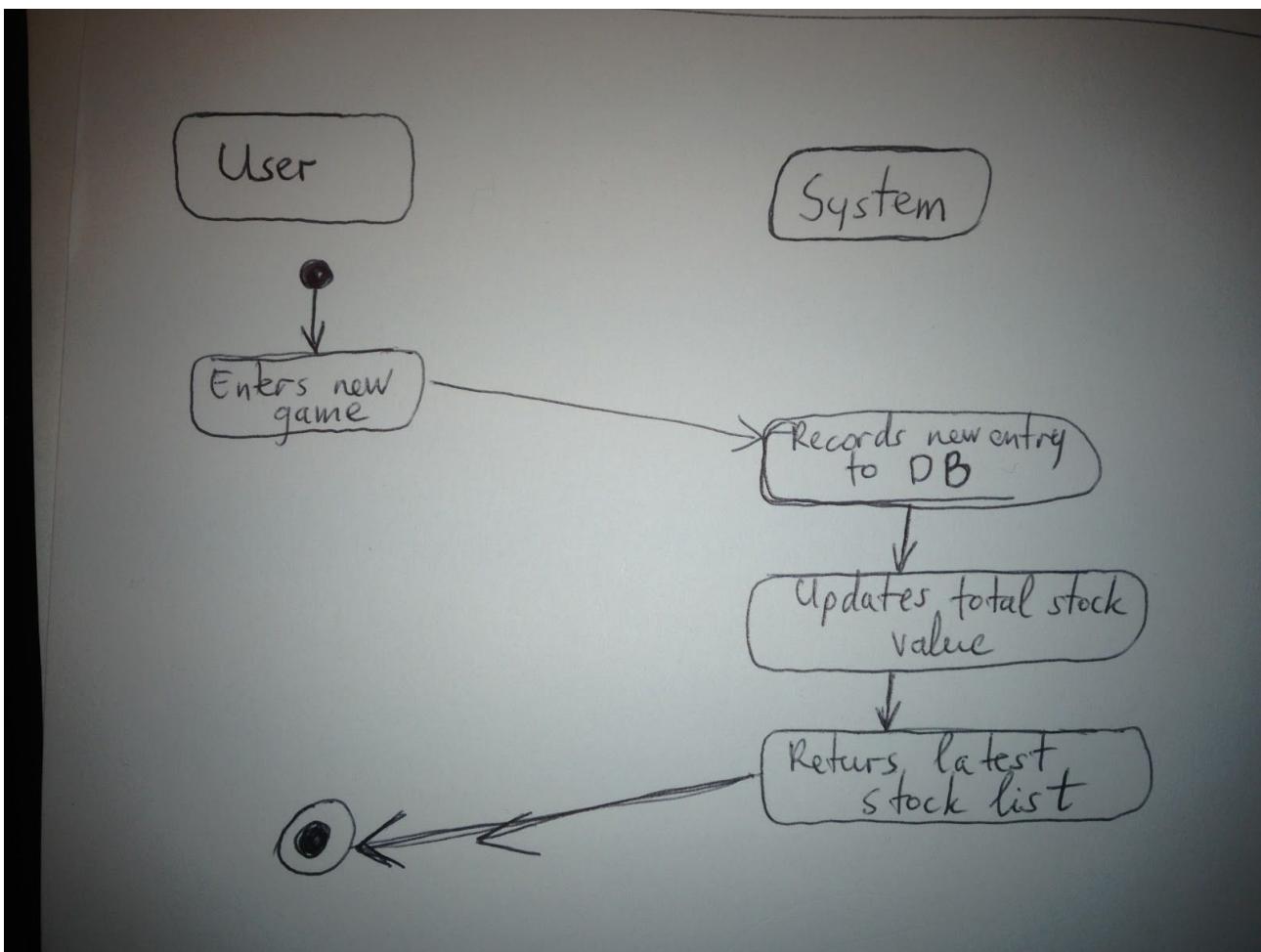
| Unit | Ref | Evidence | |
|------|-------|---------------------|--|
| A&D | A.D.3 | An Object Diagram | |
| | | Description: | |



Description:

Object diagram showing instances of developer, publisher, game, gametag, and genre classes with their attributes.

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

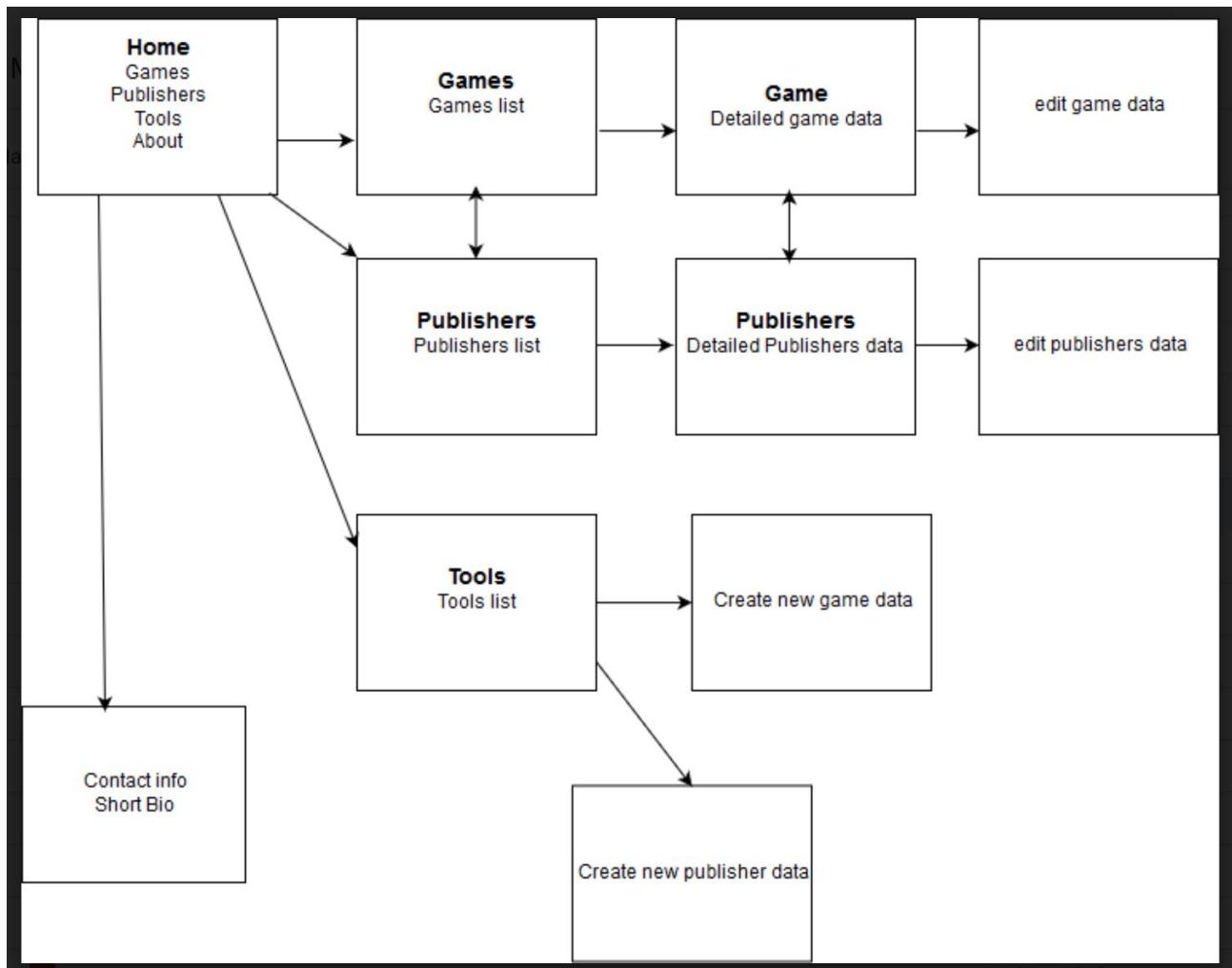


Description: Activity diagram for my Game store app, showing user activity of adding a new stock item to the store stock.

| 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: | |

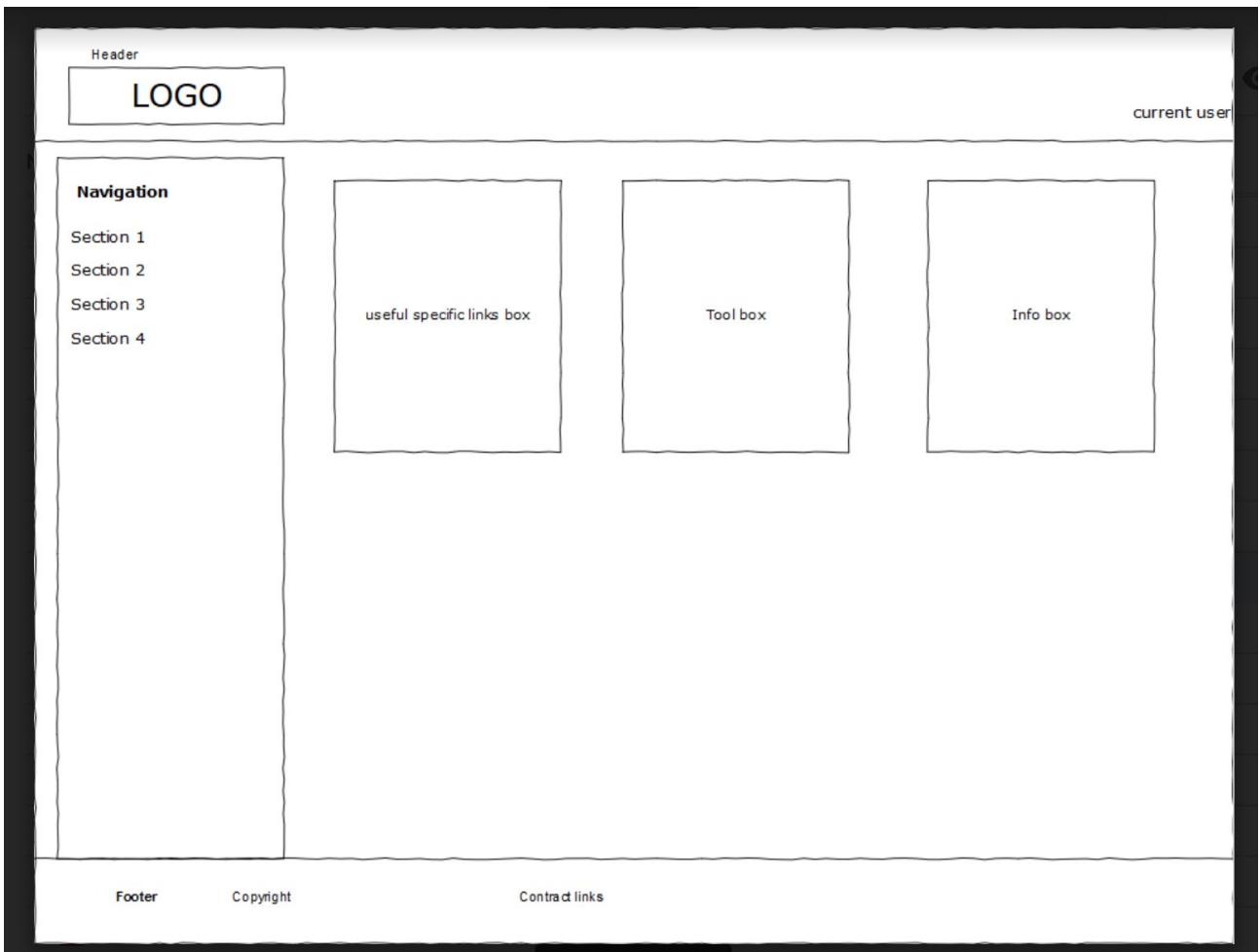
| <i>Constraint Category</i> | <i>Implementation Constraint</i> | <i>Solution</i> |
|--|--|---|
| Hardware and Software Platforms | The App is built and tested using Chrome browser, because of its superior developer tools compared to other browsers. Which leads app behaving in unexpected ways on different browsers and might be unusable for users with different browsers. | To improve apps performance on different browsers, we will set up testing environment for each different browser. So that we are aware which parts of code act differently on different browsers. |
| Performance Requirements | Most of the stock information for the app comes from external API and it needs to be updated often, so its slowing down overall performance. | To make app work faster, we will store as much information as possible in local database and reduce apps calls to external API, for example there is no need to update stock price on weekends. |
| Persistent Storage and Transactions | All of user data is stored on a physical disk, which have a risk of being lost. | To secure user data we will move our database to the cloud storage. |
| Usability | Since the app is geared for a small volume stock trader, so user may not be well informed on details of stock market, or have limited ability to read graph. | To help user to get the most information and use from the app, we will detailed explanation and guides to understanding graphs and suggestions for the future. |
| Budget | Since budget is extremely limited, we are limited to a few developers, with no possibility to adding new people or getting extra software. | To mitigate budget constraints we will use best available open source software and use or the help Codeclan provides. |
| Time | Time is strictly limited to 6 days, which reduces of amount of possible features. | To ensure that the app is functional and attractive to the user, we will work on the most vital bits of the app. |

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

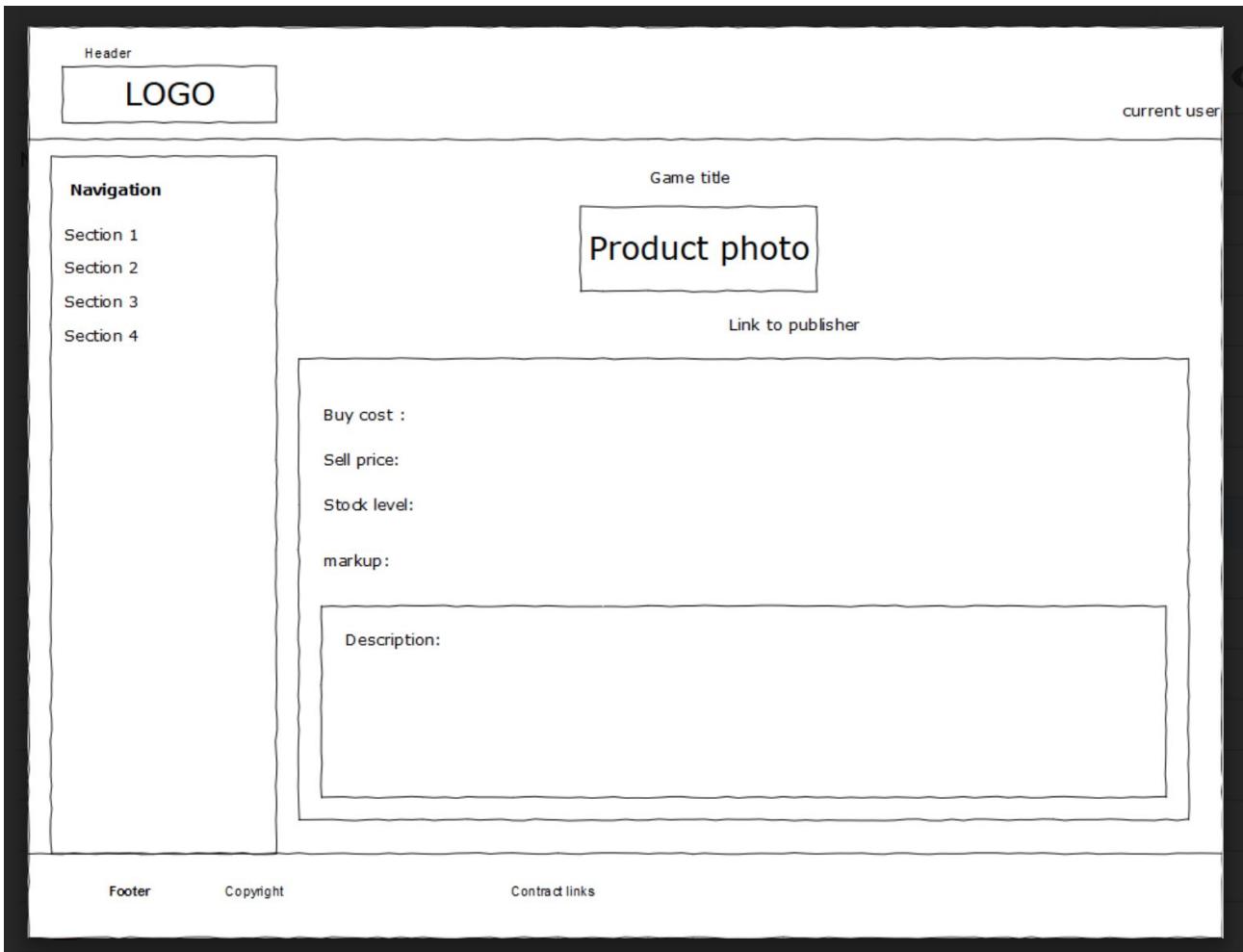


Description:
Site map showing available paths in the game store app.

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



Description:
Game-store app's homepage wireframe.



Description:

Game-store app's wireframe for individual product page.

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

```

PrimeChecker.prototype.bindEvents = function () {
    // listen for when form is submitted and published
    // create new variable from number received
    // run function to check if number is prime
    // publish result in result-calculated;

    PubSub.subscribe('FormView:number-submitted', (event) => {
        const inputtedNumber = event.detail;
        const result = this.isNumberPrime(inputtedNumber);
        PubSub.publish('PrimeChecker:result-calculated', result);
    });
};

```

Description here: Pseudo code for the bind events function, which lets PubSub model to work.

| 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: | |

Create new developer

[Home](#)

[Games](#)

[Developers](#)

[Publishers](#)

[Tools](#)

[About](#)

[Contact](#)

Name:

Warhorse Studios

Location:

Czech Republic

Logo image link:

[/Warhorse_Studios.png](#)

[Create developer](#)

Developers

Home

Games

Developers

Publishers

Tools

About

Contact

Name

Location

[Bethesda Game Studios](#)

| USA

[BioWare](#)

| Canada

[Blue Byte](#)

| Germany

[CD Projekt Red](#)

| Poland

[Colossal Order](#)

| Finland

[Crystal Dynamics](#)

| USA

[Dontnod Entertainment](#)

| France

[Eidos Montréal](#)

| Canada

[Firaxis Games](#)

| USA

[Frozenbyte](#)

| Finland

[Haemimont Games](#)

| Bulgaria

[Obsidian](#)

| USA

[Rockstar North](#)

| Scotland

[Ubisoft Montreal](#)

| Canada

[Ubisoft Paris](#)

| France

[Valve Corporation](#)

| USA

[Warhorse Studios](#)

| Czech Republic

Create new developer

[Home](#)[Games](#)[Developers](#)[Publishers](#)[Tools](#)[About](#)[Contact](#)

Developed games:

[EDIT DEVELOPER](#)[delete developer](#)

Description:

User inputs data for new game developer, after pressing create button new developer appears in developers list. Separate page for this developer also created.

| 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: | |

Create new publisher

Name:**Location:**[Create publisher](#)

Add new game:

Name:

Developed by:

Published by:

- ✓ 2K Games
- Atari
- Bethesda Softworks
- CD Projekt
- Deep Silver**
- Electronic Arts
- Kalypso Media
- Nobilis
- Paradox Interactive
- Rockstar Games
- Square Enix
- Ubisoft
- Valve Corporation

Selling Price:

Select tags:

- RPG
- Action
- Adventure



Kingdom Come: Deliverance

Developed by: Warhorse Studios

Published by: Deep Silver

Released: 2018

Description:

An action role-playing game set in an open world environment and played from a first-person perspective which utilizes a classless RPG system, allowing the player to customise their skills to take on roles such as a warrior, bard, thief or their hybrids.

Genres: RPG, Adventure, Open-World [Edit tags](#)

- Buying cost: 35
- Selling price: 45
- Markup: 10
- In stock: 5 low stock

[EDIT THIS GAME](#)

[delete this game](#)

Description:

User inputs data for a new publisher, publisher is saved and then the user is able to select publisher from a dropdown menu whilst adding a new game to inventory. After the game is added, publishers name will be displayed on the game page as a link, which could take user to the publishers page where all the that was inputted in the beginning will be available.

| 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: |

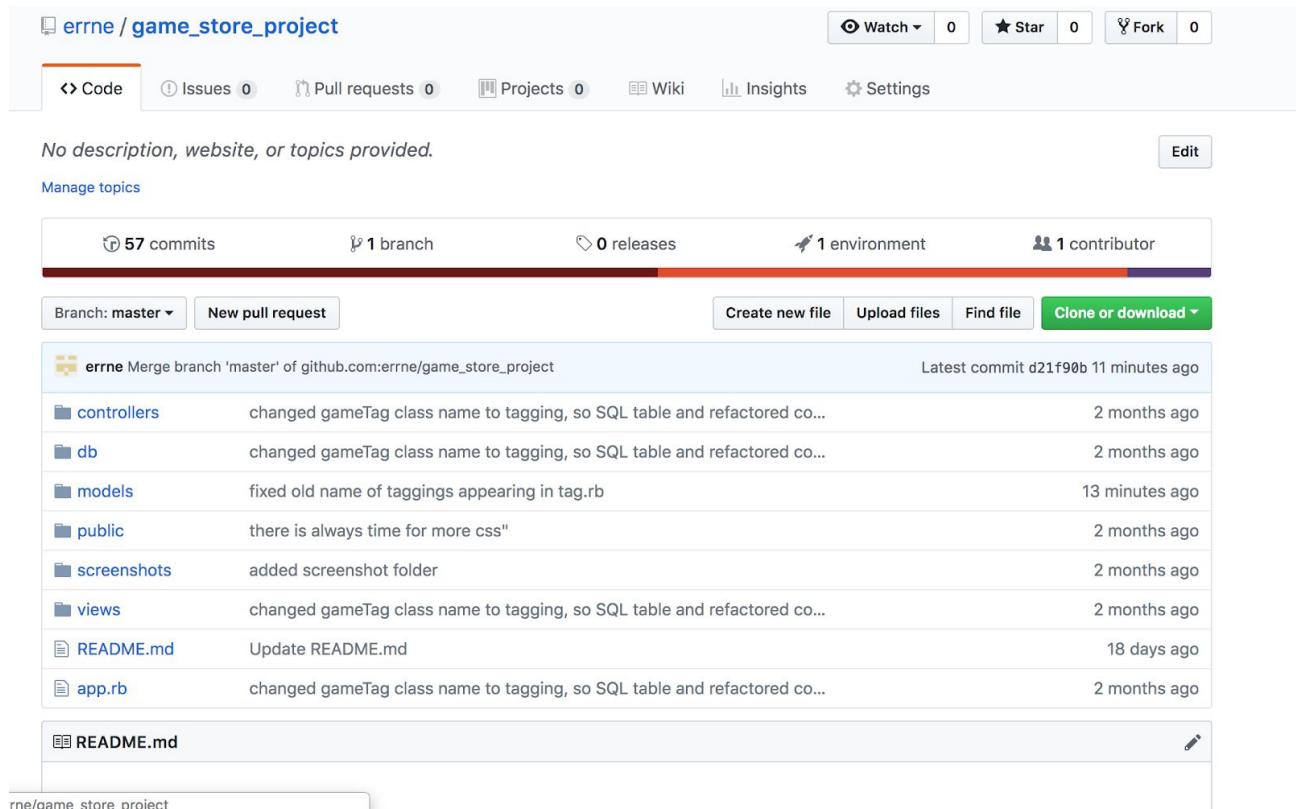
| Games | | | | | | | | | |
|---------|---------------------------|-----------------------|---------------------|--------------|----------|------|---------------|----------------------------|--|
| | Name | Developer | Publisher | Release Year | In Stock | Cost | Selling Price | Tags | |
| | Assassin's Creed II | Ubisoft Montreal | Ubisoft | 2009 | 9 | 9 | 15 | Adventure, Action | |
| Cities: | Cities: Skylines | Colossal Order | Paradox Interactive | 2015 | 11 | 12 | 21 | City-builder | |
| | Civilization V | Firaxis Games | 2K Games | 2010 | 11 | 12 | 19 | City-builder | |
| | Deus Ex: Human Revolution | Eidos Montréal | Square Enix | 2011 | 7 | 12 | 22 | FPS, RPG | |
| | Dragon Age II | BioWare | Electronic Arts | 2011 | 7 | 11 | 24 | RPG | |
| | Dragon Age: Origins | BioWare | Electronic Arts | 2009 | 7 | 11 | 19 | RPG, Adventure | |
| | Elder Scrolls: Morrowind | Bethesda Game Studios | Bethesda Softworks | 2002 | 7 | 5 | 15 | RPG, Adventure, Open-World | |
| | Elder Scrolls: Oblivion | Bethesda Game Studios | Bethesda Softworks | 2006 | 3 | 9 | 19 | RPG, Adventure, Open-World | |
| | Elder Scrolls: Skyrim | Bethesda Game Studios | Bethesda Softworks | 2011 | 9 | 17 | 25 | RPG, Adventure, Open-World | |
| | Fallout 3 | Bethesda Game Studios | Bethesda Softworks | 2008 | 9 | 6 | 12 | Open-World, FPS | |
| | Fallout: New Vegas | Obsidian | Bethesda Softworks | 2010 | 8 | 11 | 20 | RPG, Open-World | |

User is selecting with a mouse-cursor “city-builder” tag in the list of all the games.

| Games | | | | | | | | | |
|-------|------------------|-----------------|---------------------|--------------|----------|------|---------------|--------------|--|
| | Name | Developer | Publisher | Release Year | In Stock | Cost | Selling Price | Tags | |
| | Cities: Skylines | Colossal Order | Paradox Interactive | 2015 | 11 | 12 | 21 | City-builder | |
| | Tropico 4 | Haemimont Games | Kalypso Media | 2011 | 3 | 13 | 25 | City-builder | |
| | Civilization V | Firaxis Games | 2K Games | 2010 | 11 | 12 | 19 | City-builder | |

Previous action has filtered the list to only display the games with the selected tag.

| 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: | |

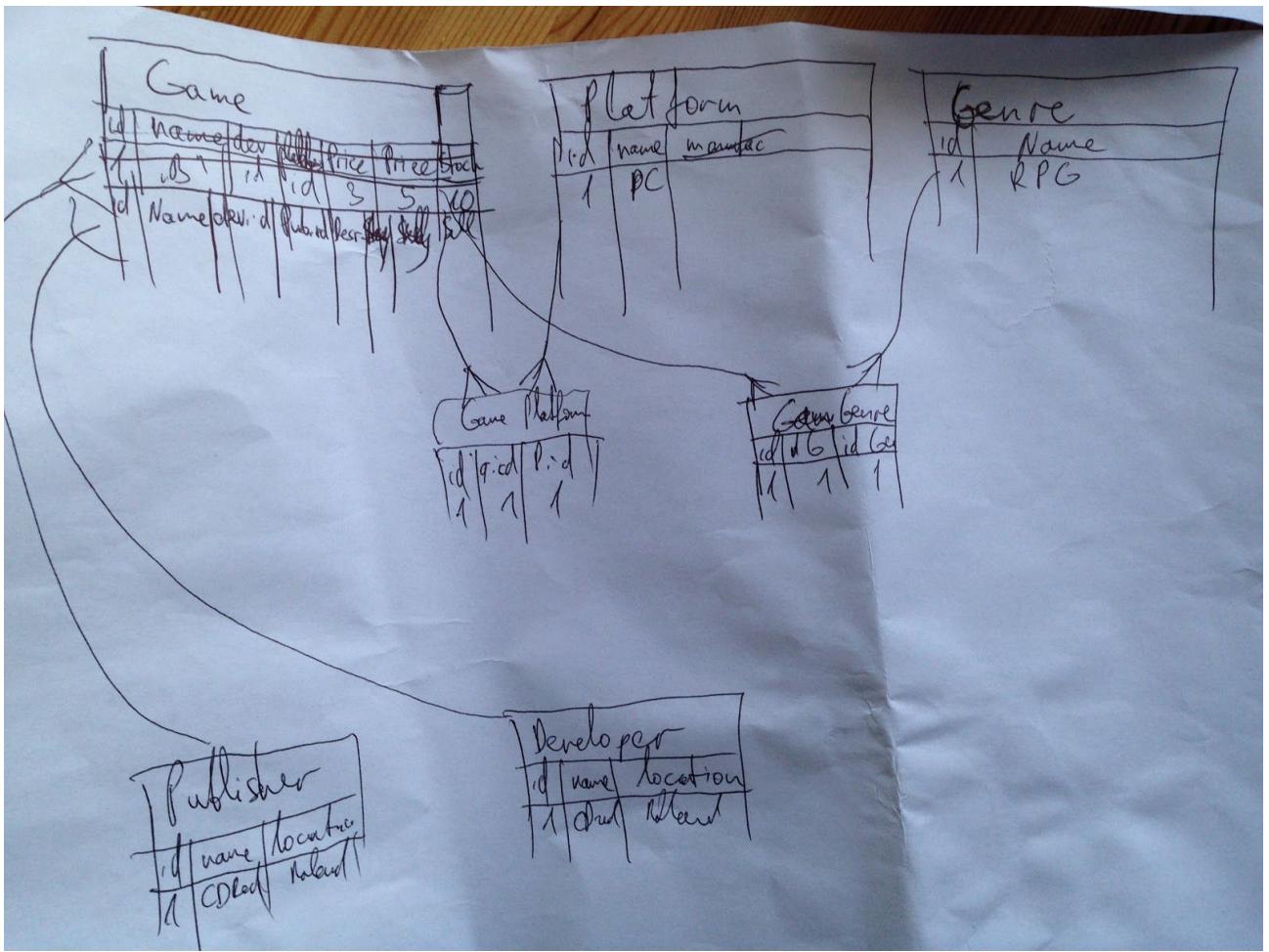
A screenshot of a GitHub repository page for 'errne / game_store_project'. The page shows basic statistics: 57 commits, 1 branch, 0 releases, 1 environment, and 1 contributor. A list of commits is displayed, all made by 'errne' to the 'master' branch. The commits involve refactoring class names from 'gameTag' to 'tagging' across various files like controllers, db, models, public, screenshots, views, README.md, and app.rb. The latest commit was 11 minutes ago. A 'Description' section at the bottom notes: 'No description, website, or topics provided.' and includes a link to the repository.

Description:

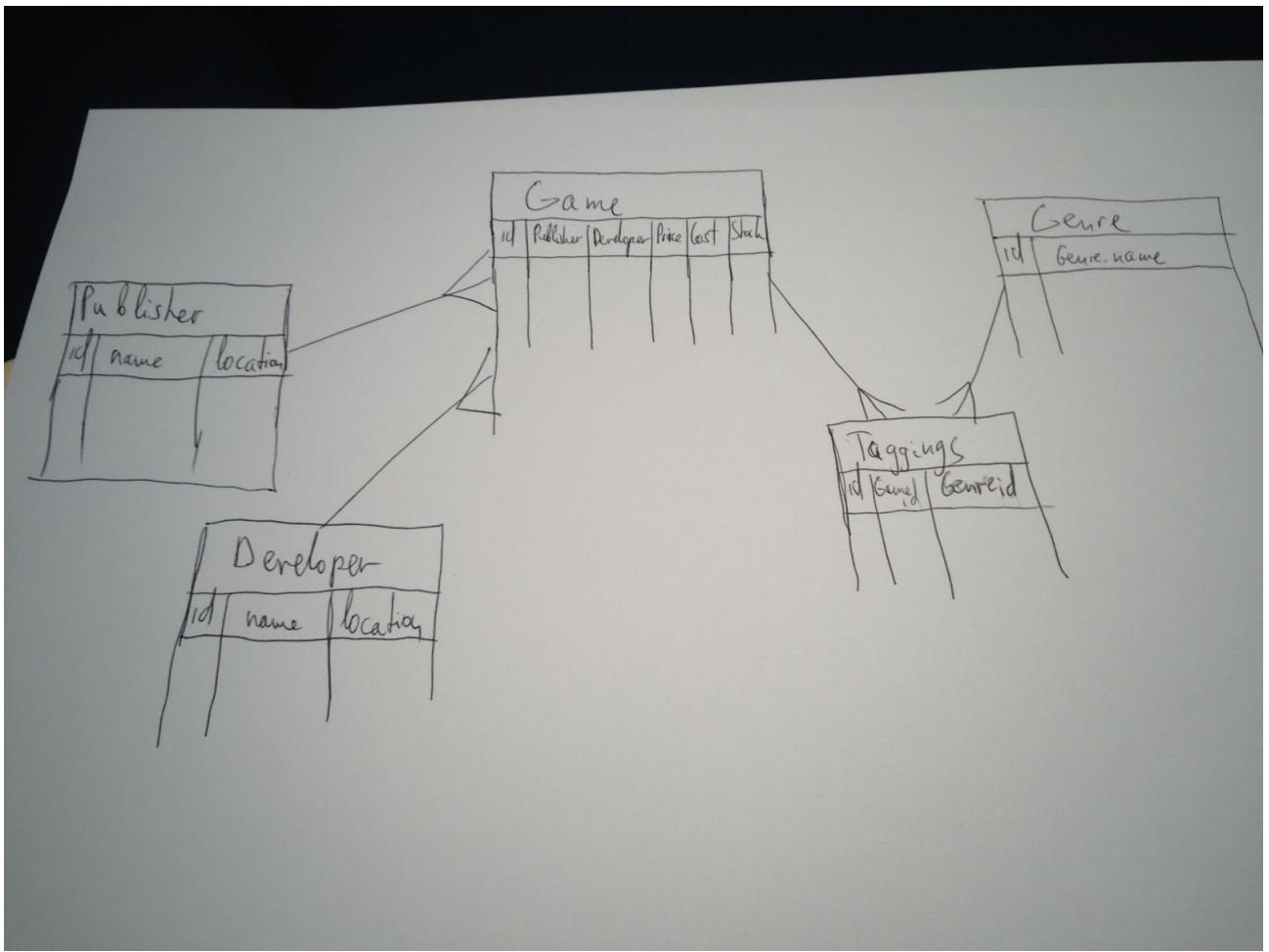
Screenshot of my solo project on GitHub.

Link: https://github.com/errne/game_store_project

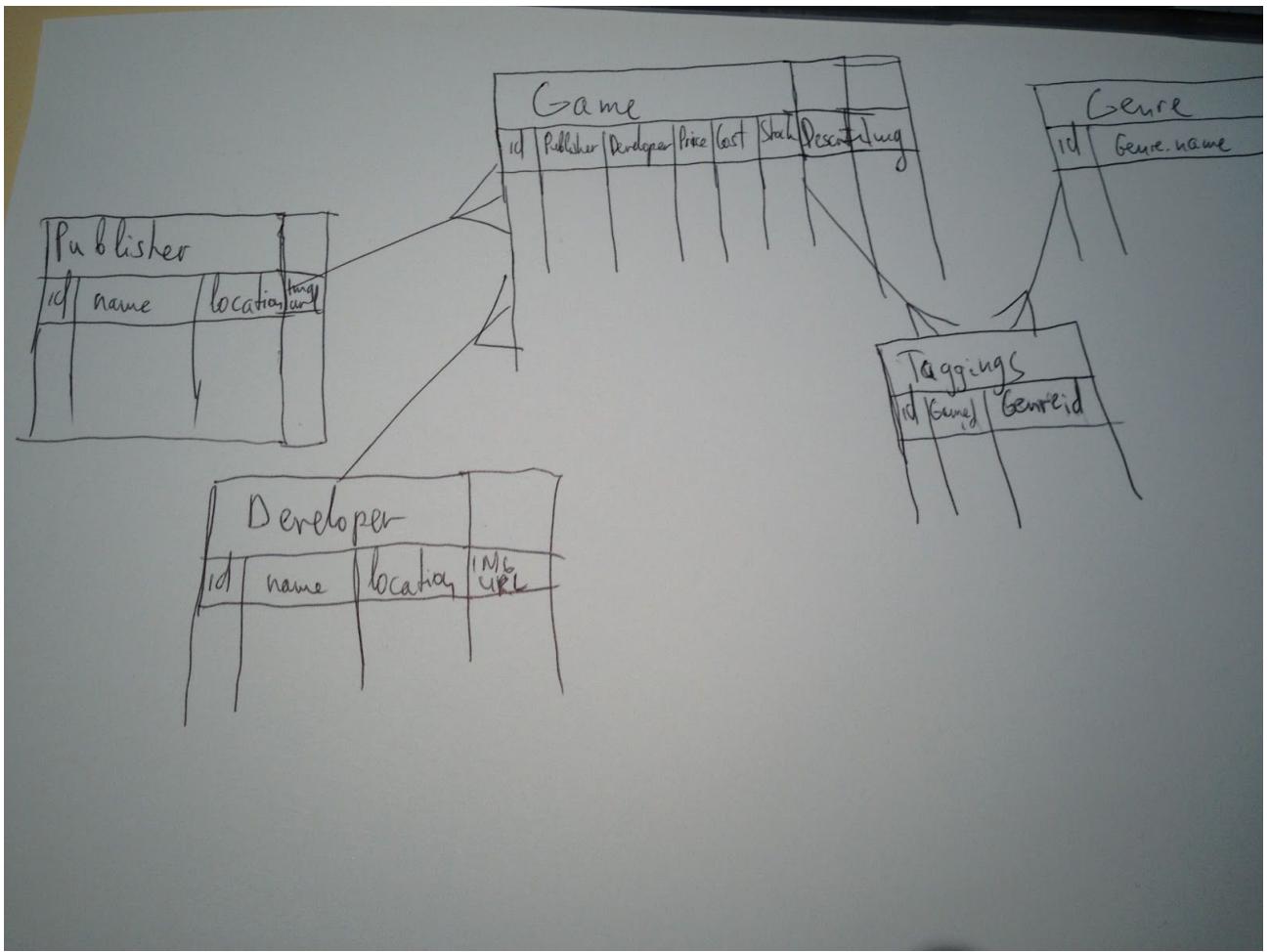
| Unit | Ref | Evidence |
|------|------|--|
| P | P.12 | Take screenshots or photos of your planning and the different stages of development to show changes. |
| | | Description: |



Description: Initial plan for solo Ruby project “Games store management”, class and data structure.



Description: Second stage of data structure, with better naming. Platform table was dropped.



Description: Final data structure for the project, publisher and developer tables got added column for image url, and game table got added image and description.

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: | |

Project

- weekend_hm_beers
 - .git
 - node_modules
 - public
 - css
 - main.css
 - js
 - index.html
 - src
 - helpers
 - models
 - beers.js
 - views
 - beer_list_view.js
 - beer_view.js
 - error_view.js
 - select_view.js
 - .DS_Store
 - app.js
 - .DS_Store
 - .gitignore
 - package-lock.json
 - package.json
 - webpack.config.js

index.html main.css beers.js

```
17
18 Beers.prototype.getData = function () {
19   const url = `https://api.punkapi.com/v2/beers?per_page=50`;
20   const request = new Request(url);
21   request.get()
22     .then((beers) => {
23       this.beerList = beers;
24       const optionsList = this.getListOfAbvs();
25       optionsList.sort(function(a,b) { return a - b;});
26       console.log(optionsList);
27       PubSub.publish('Beers:all-ready', this.beerList);
28       PubSub.publish('Beers:options-ready', optionsList)
29     })
30     .catch((err) => {
31       PubSub.publish('Beers:error', err);
32     });
33   };
34
35 Beers.prototype.getDataId = function (id) {
36   const url = `https://api.punkapi.com/v2/beers/${id}`;
37   const request = new Request(url);
38   request.get()
39     .then((beer) => {
40       this.beer = beer;
41       PubSub.publish('Beers:bearInfo-ready', this.beer);
42     })
43     .catch((err) => {
44       PubSub.publish('Beers:error', err);
45     });

```

Beer!

Select strength: 

Trashy Blonde

A titillating, neurotic, peroxide punk of a Pale Ale. Combining attitude, style, substance, and a little bit of low self esteem for good measure; what would your mother say? The seductive lure of the sassy passion fruit hop proves too much to resist. All that is even before we get onto the fact that there are no additives, preservatives, pasteurization or strings attached. All wrapped up with the customary BrewDog bite and imaginative twist.

alc/vol: 4.1

First brewed: 04/2008



Description:

First screenshot shows two functions using different API endpoints to get information about Brewdog bears.

Next screenshot shows website displaying information retrieved from Brewdog API and filtered by ABV.

| Unit | Ref | Evidence |
|------|------|---|
| P | P.18 | <p>Demonstrate testing in your program. Take screenshots of:</p> <ul style="list-style-type: none"> * Example of test code * The test code failing to pass * Example of the test code once errors have been corrected * The test code passing |
| | | Description: |

```

Atom  File  Edit  View  Selection  Find  Packages  Window  Help
PDA — user@users-MacBook-Pro — ~/CodeClan/PDA
from PDA_Static_and_Dynamic_Task_A/card_g
PDA ruby PDA_Static_and_Dynamic_Task_A/card_g
PDA_Static_and_Dynamic_Task_A/card_game_spec.rb:3
user/CodeClan/PDA/PDA_Static_and_Dynamic_Task_A/c
unexpected end-of-input, expecting keyword_end (Sy
from PDA_Static_and_Dynamic_Task_A/card_g
PDA ruby PDA_Static_and_Dynamic_Task_A/card_g
Run options: --seed 60095

# Running:

E

Finished in 0.001006s, 994.0358 runs/s, 0.0000 as

1) Error:
CardGameTest#test_card_game_can_check_for_ace:
NoMethodError: undefined method `checkforAce' for
"diamond", @value=>
  PDA_Static_and_Dynamic_Task_A/card_game_spec.
ck_for_ace'

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

```

Project

- PDA
 - arrhash
 - PDA_Static_and_Dynamic_Task_A
 - card_game_spec.rb
 - card_game.rb
 - card.rb
 - testing_task_1.md
 - .DS_Store

card_game_spec.rb — ~/CodeClan/PDA

```

card_game_spec.rb — ~/CodeClan/PDA
testing_task_1.md card.rb card_game.rb card_game_spec.rt ×
1 require('minitest/autorun')
2 require('minitest/rails')
3 require_relative('card_game.rb')
4
5 class CardGameTest < MiniTest::Test
6
7   def test_card_game_can_check_for_ace
8     card1 = Card.new("diamond", 1)
9     assert_equal(true, card1.checkforAce(card1))
10   end
11
12
13 end
14

```

PDA_Static_and_Dynamic_Task_A/card_game_spec.rb 8:35

LF UTF-8 Ruby 0 files

```

PDA_Static_and_Dynamic_Task_A git:(master) ✘ ruby card_game_spec.rb
Run options: --seed 11939

# Running:
E..EE.

Finished in 0.001222s, 4909.9837 runs/s, 2454.9918 assertions/s.

 1) Error:
CardGameTest#test_card_game_returns_higher_card_equal_value:
NameError: undefined local variable or method `card' for #<CardGame:0x007f91b39495e8>
Did you mean? card2
    card1
 /Users/user/CodeClean/PDA/ernest_fraiman_pda/PDA_Static_and_Dynamic_Task_A/card_game.r
b:14:in `highest_card'
 card_game_spec.rb:37:in `test_card_game_returns_higher_card_equal_value'

 2) Error:
CardGameTest#test_card_game_returns_higher_card:
NameError: undefined local variable or method `card' for #<CardGame:0x007f91b3948148>
Did you mean? card2
    card1
 /Users/user/CodeClean/PDA/ernest_fraiman_pda/PDA_Static_and_Dynamic_Task_A/card_game.r
b:14:in `highest_card'
 card_game_spec.rb:23:in `test_card_game_returns_higher_card'

 3) Error:
CardGameTest#test_card_game_returns_higher_card_card1_higher:
NameError: undefined local variable or method `card' for #<CardGame:0x007f91b38e3180>
Did you mean? card2
    card1
 /Users/user/CodeClean/PDA/ernest_fraiman_pda/PDA_Static_and_Dynamic_Task_A/card_game.r
b:14:in `highest_card'
 card_game_spec.rb:30:in `test_card_game_returns_higher_card_card1_higher'

6 runs, 3 assertions, 0 failures, 3 errors, 0 skips

```

Description: Two above screenshots, show test code and failing test run. On the right side of the screenshots you can see test code for the card game. On the left side test running and failing in the terminal window.

```

Compressing objects: 100% (1/1), done.
Writing objects: 100% (2/2), 227 bytes | 227.00 KiB/s, done.
Total 2 (delta 0), reused 1 (delta 0)
To github.com:ernne/ernest_fraiman_pda.git
 * [new branch] master -> master
→ ernest_fraiman_pda git:(master) mkdir PDA_Static_and_Dynamic_Task_A
→ ernest_fraiman_pda git:(master) git add .
→ ernest_fraiman_pda git:(master) × git commit -m "repushing testing task"
[master 1cab9bb] repushing testing task
 4 files changed, 135 insertions(+)
 create mode 100444 PDA_Static_and_Dynamic_Task_A/card.rb
 create mode 100444 PDA_Static_and_Dynamic_Task_A/card.game.rb
 create mode 100444 PDA_Static_and_Dynamic_Task_A/card.game_spec.rb
 create mode 100444 PDA_Static_and_Dynamic_Task_A/testing_task_1.md
→ ernest_fraiman_pda git:(master) git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.73 KiB | 1.73 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To github.com:ernne/ernest_fraiman_pda.git
 e78522...1cab9bb master -> master
→ ernest_fraiman_pda git:(master) atom
→ ernest_fraiman_pda git:(master) git add .
→ ernest_fraiman_pda git:(master) × git commit -m "improved comments on static test task"
[master 9bc351e] improved comments on static test task
 1 file changed, 12 insertions(+), 10 deletions(-)
→ ernest_fraiman_pda git:(master) git push
zsh: command not found: git
→ ernest_fraiman_pda git:(master) git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 784 bytes | 704.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:ernne/ernest_fraiman_pda.git
 1cab9bb...9bc351e master -> master
→ ernest_fraiman_pda git:(master) cd PDA_Static_and_Dynamic_Task_A
→ PDA_Static_and_Dynamic_Task_A git:(master) ls
card.rb          card.game.rb      card.game_spec.rb  testing_task_1.md
→ PDA_Static_and_Dynamic_Task_A git:(master) mkdir screenshots
→ PDA_Static_and_Dynamic_Task_A git:(master) ruby card_game_spec.rb
Run options: --seed 5824

# Running:
[.....]

Finished in 0.001074s, 5586.5923 runs/s, 5586.5923 assertions/s.

6 runs, 6 assertions, 0 failures, 0 errors, 0 skips

```

Description: Screenshot of fixed code and successful tests. On the right side of the screen corrected test code, on the left the test code passing.

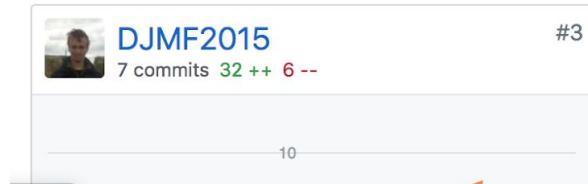
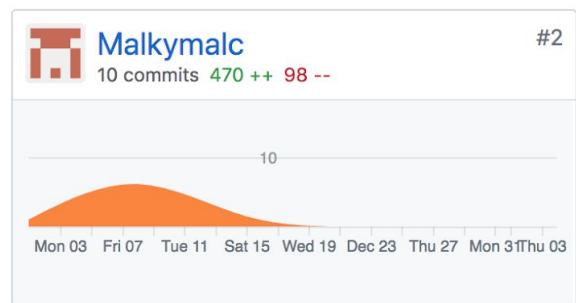
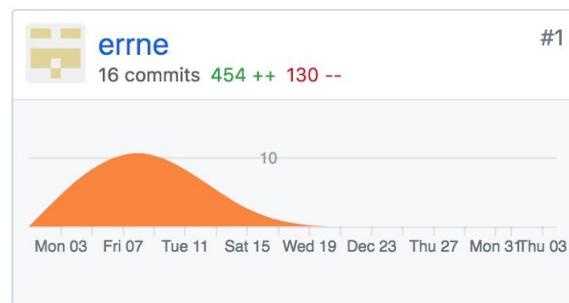
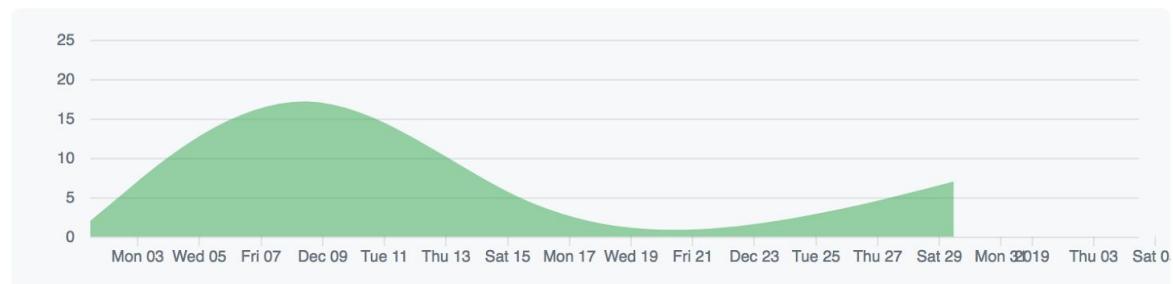
Week 9

| 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. | |
| | | Description: | |

Dec 2, 2018 – Jan 5, 2019

Contributions: Commits ▾

Contributions to master, excluding merge commits



Description:

Contributor's page from javascript group group project 'shares portfolio'. (errne is my github account name)

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

Restaurant Booking System:

You have been tasked to create a booking system for a brand new restaurant. The restaurant needs a way to book and arrange tables for customers who are booking over the phone. This system is for the staff to use.

MVP:

Your system must be able to:

- Allow a customer to book a table at the restaurant for a particular time and date
- Update a booking, for example if the customer wants to change a booking time
- Display a list of bookings for a given date
- Display a list of customers ordered by frequency of visits

Project Extensions:

- Don't allow double bookings
- Add a customer's receipt to a booking so you can view their previous orders and how much they spent
- Calculate how much a customer has spent over a given period of time
- Give discounts to frequent customers
- Whatever features you think would be beneficial to a restaurant

Description: Brief for a full stack group project I was part of.

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

CLIENT

- PUBLIC

- CSS FOLDER
- JS FOLDER
- index.html

- SRC

- HELPERS FOLDER

- ↳ pubsub.js
- ↳ request-helper.js

- MODELS FOLDER

- ↳ stocks.js
- ↳ charts.js

- VIEWS FOLDER

- ↳ stock-form-view
- ↳ stock-list-view
- ↳ portfolio-list-view
- ↳ stock-detail-view
- ↳ list-item-view

- app.js file

SERVER

- DB

- seeds.js

- HELPERS

- create-router.js

- Server

- webpack.config.js

- .gitignore

- ↳ inside write

- .node_modules

- .bundle.js



PROJECT - SHARES PORTFOLIO

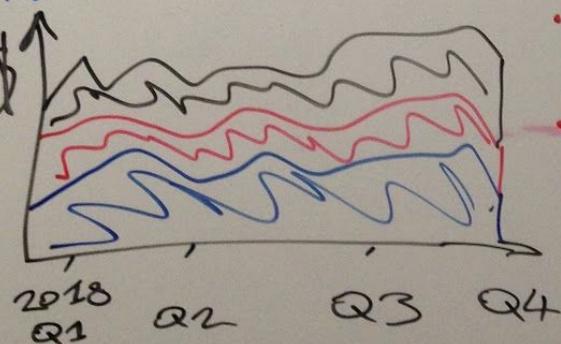
MVP

- ✓① View total current value
- ✓② View individual + total performance trends
- ✓③ Retrieve a list of share prices from an external A.P.I + allow the user to add shares to her portfolio
- ✓④ View a chart of current values in her portfolio.

Tools

- Mongo
- info from API
+ put in her portfolio

Markets?



Charting
Library?
chart.js

USER NEEDS

- be able to analysis her existing portfolio
- View all her shares
- Know how much all shares are worth.
- View them all seperately.
- be able to add new shares to her portfolio

Things To Do

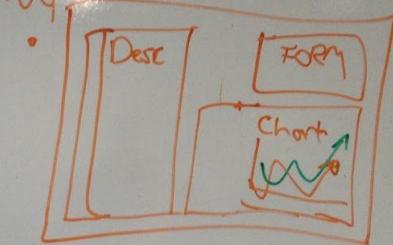
- Total Portfolio Value
 - Edit [Add to Portfolio] button.
 - Render ^{Pie} Chart: - Ernest
 - Round \$ Amounts
- MVP Extensions Bokah
- Format Change displays [Port list, stock list, tabular summary] for + all - (green) (red)
 - Add ^{Hi ?} day low + info to sidebar & mobile nav

Malcolm.

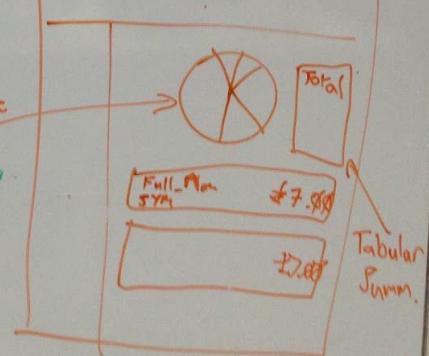
[3:30 pm]

12/12/18

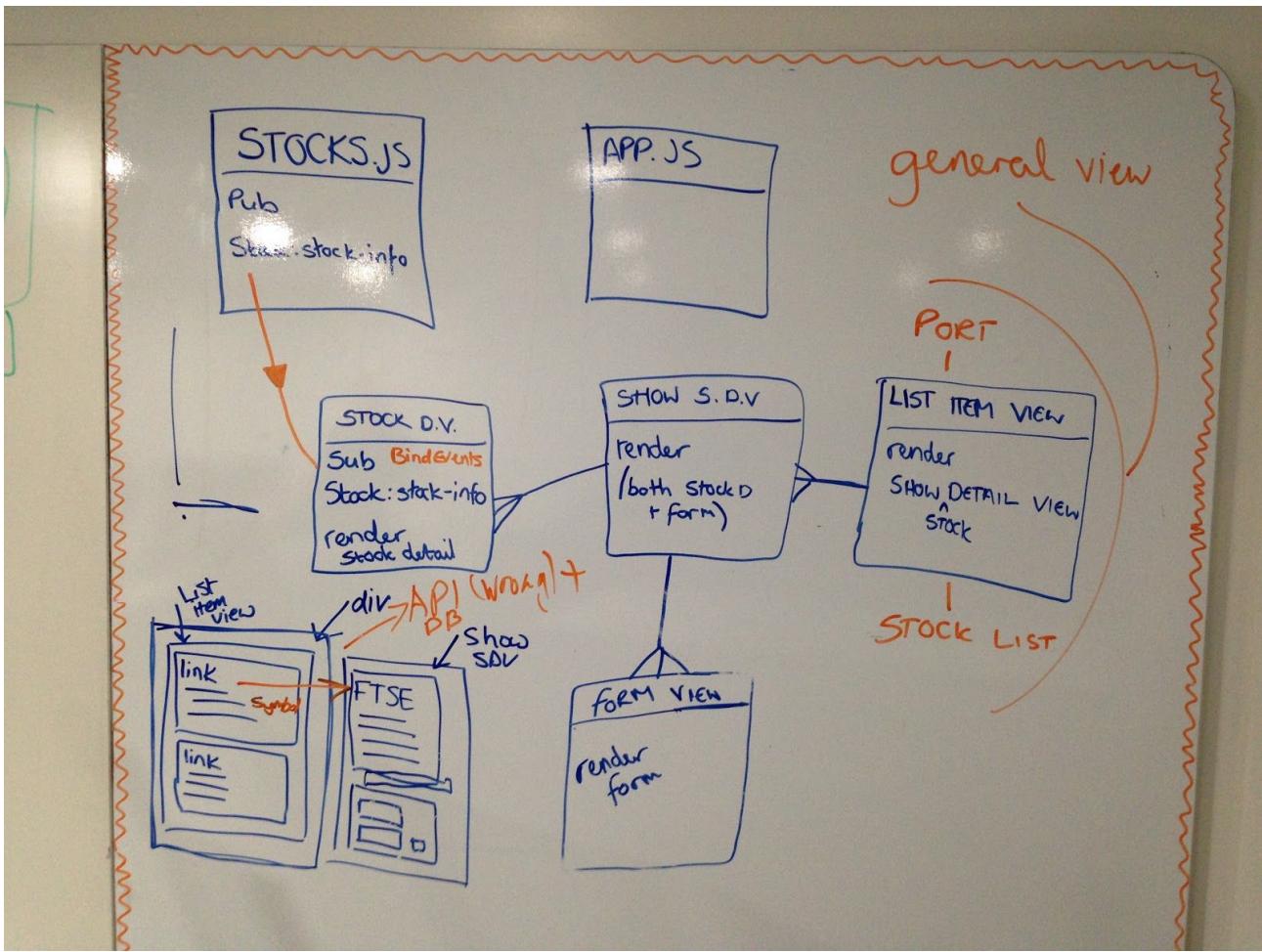
\$7.686666666666666 charts



Graphic summ.



Line Charts



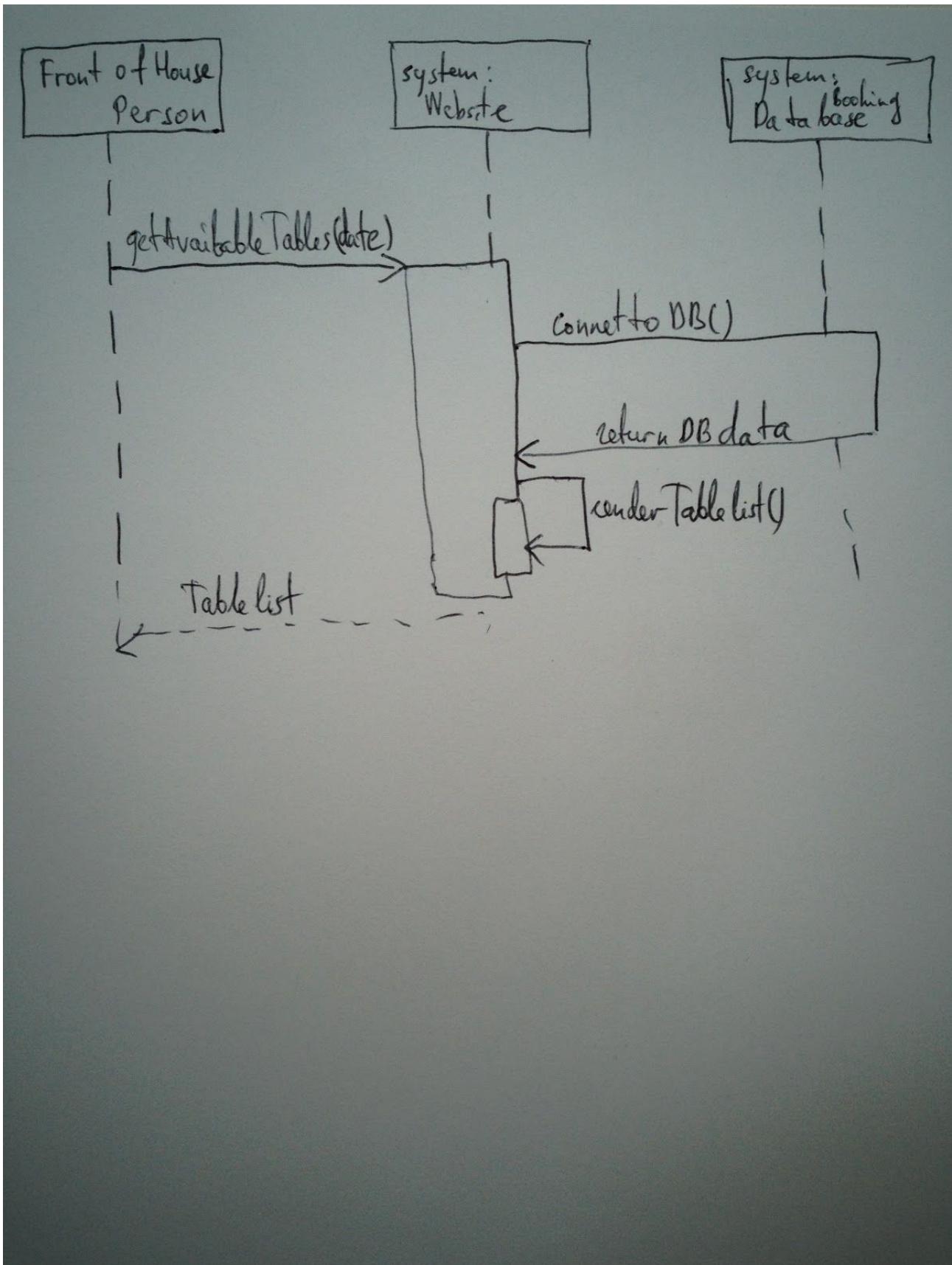
Description: Photos of planning we did for group Javascript project “Shares Portfolio”, most of the planning was done on the whiteboards.

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

| <u>Acceptance Criteria</u> | <u>Expected Output</u> | <u>Pass/Fail</u> |
|---------------------------------------|--|-------------------------|
| See all the games in the stock | Selecting ‘Games’ option, shows stock list of all available games. | <u>Pass</u> |
| Add new game to the stock | Pressing “Add new game” button, opens up a form. Which when filled add new game to the stock. | <u>Pass</u> |
| See all the games of particular genre | Pressing of a genre name on stock list filters the list, only to show games of selected genre. | <u>Pass</u> |

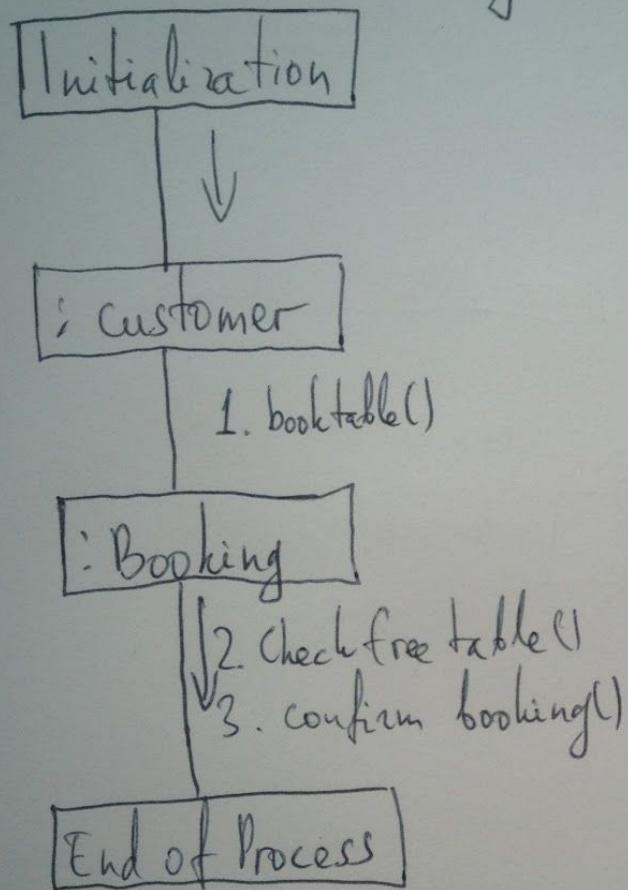
Description: Some of acceptance criteria for game store owner, who wants to manage the stock of the store.

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



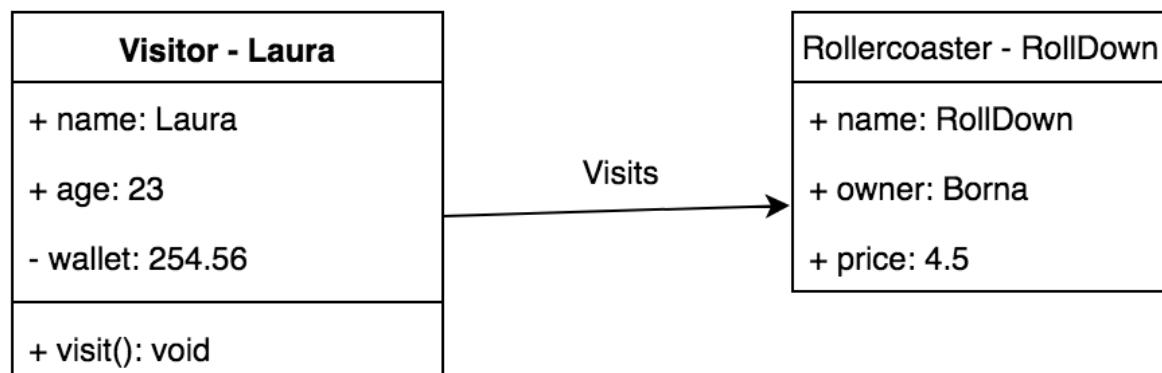
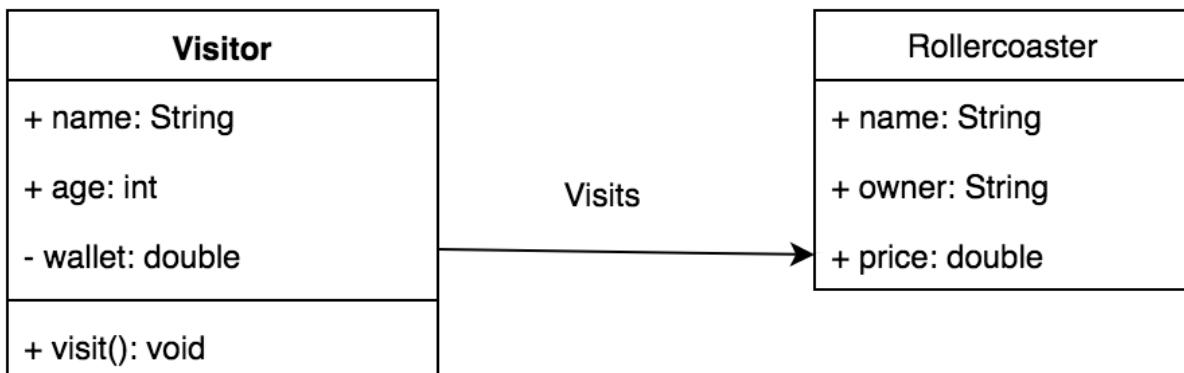
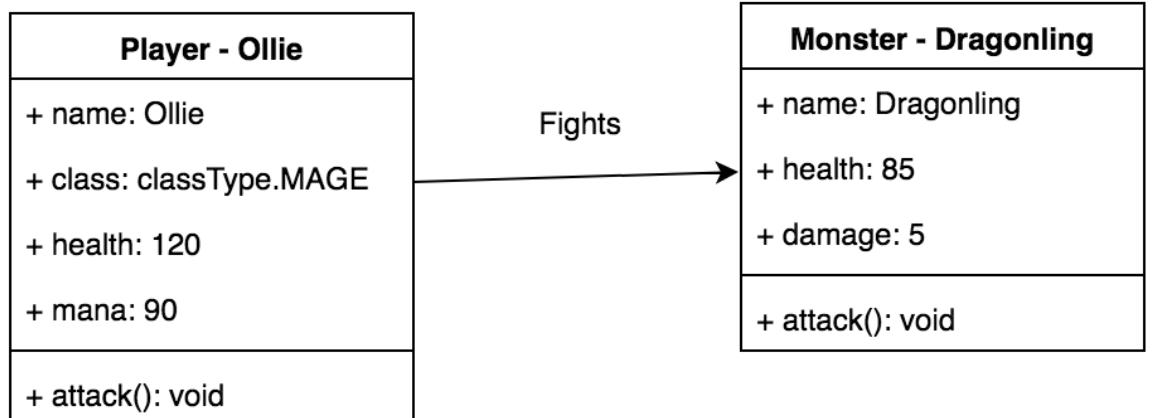
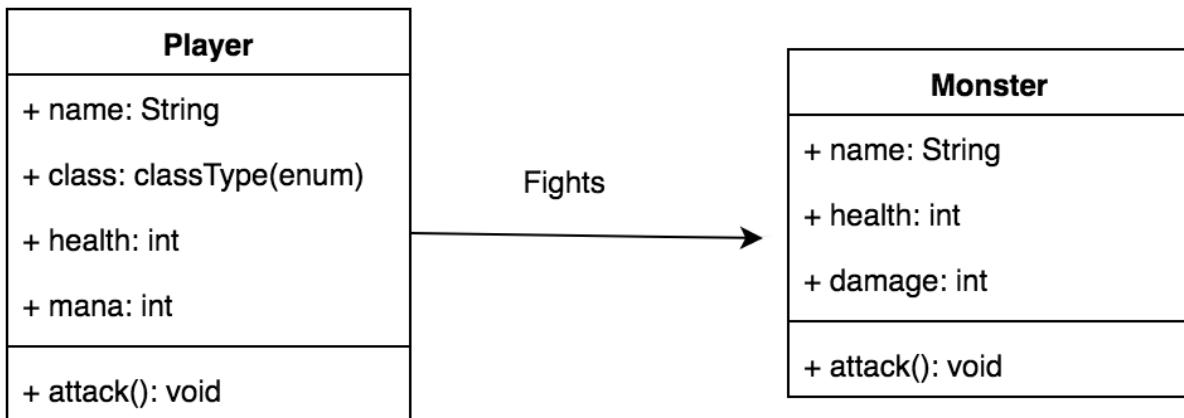
Description: Sequence diagram for Restaurant Booking app.

Restaurant Booking System Collaboration Diagram



Description: for customer booking a table at a restaurant, for Restaurant Booking system project.

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



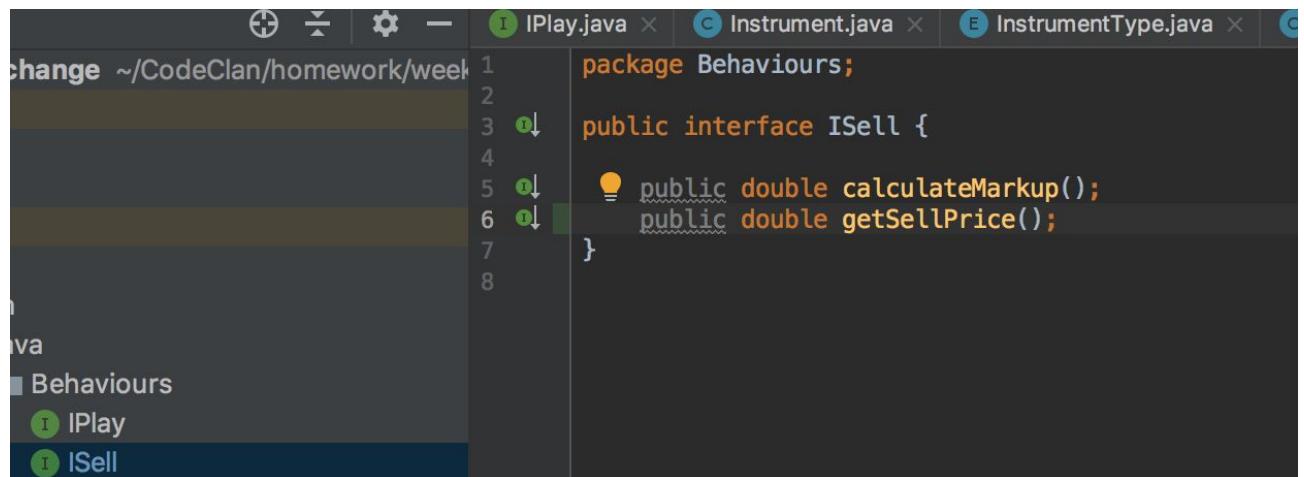
Description here

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

| Bug\Error | Solution | Date |
|---|---|-------------------|
| Available spaces in restaurant displayed incorrectly, because some bookings have negative guest. | To prevent this error, booking form input field for guest set to be equal to at least 1. | <u>06/02/2019</u> |
| Unable to filter bookings by date in the database. | Date object needs to be formatted to 'SimpleDateFormat' before sorting or filtering. | <u>06/02/2019</u> |
| While trying get the sum of the order, incorrect total is returned(22 instead of 4). | Numbers from database get returned as Strings. So they need to be parsed to integers before adding. | <u>06/02/2019</u> |
| While trying to add an order to a booking, getting error: <i>"Unhandled Rejection (TypeError): Cannot read property 'orderedItems' of undefined"</i> | function handleOrderedItemPost, which handles orders, needs to be bind inside of constructor. | <u>07/02/2019</u> |
| Some website components styling is not applied correctly | In React-JSX styling sometimes will fail, as the class keyword is reserved by ES6. Instead use className for HTML components inside React. | <u>07/02/2019</u> |

Week 12

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



```
change ~/CodeClan/homework/week1
1 package Behaviours;
2
3 public interface ISell {
4
5     public double calculateMarkup();
6     public double getSellPrice();
7 }
8
```

Behaviours

- IPlay
- ISell

```
import Behaviours.ISell;

public class StockItem implements ISell {
    private String name;
    private double buyCost;
    private double sellPrice;

    public StockItem(String name, double buyCost) {
        this.name = name;
        this.buyCost = buyCost;
        this.sellPrice = buyCost * 1.25;
    }

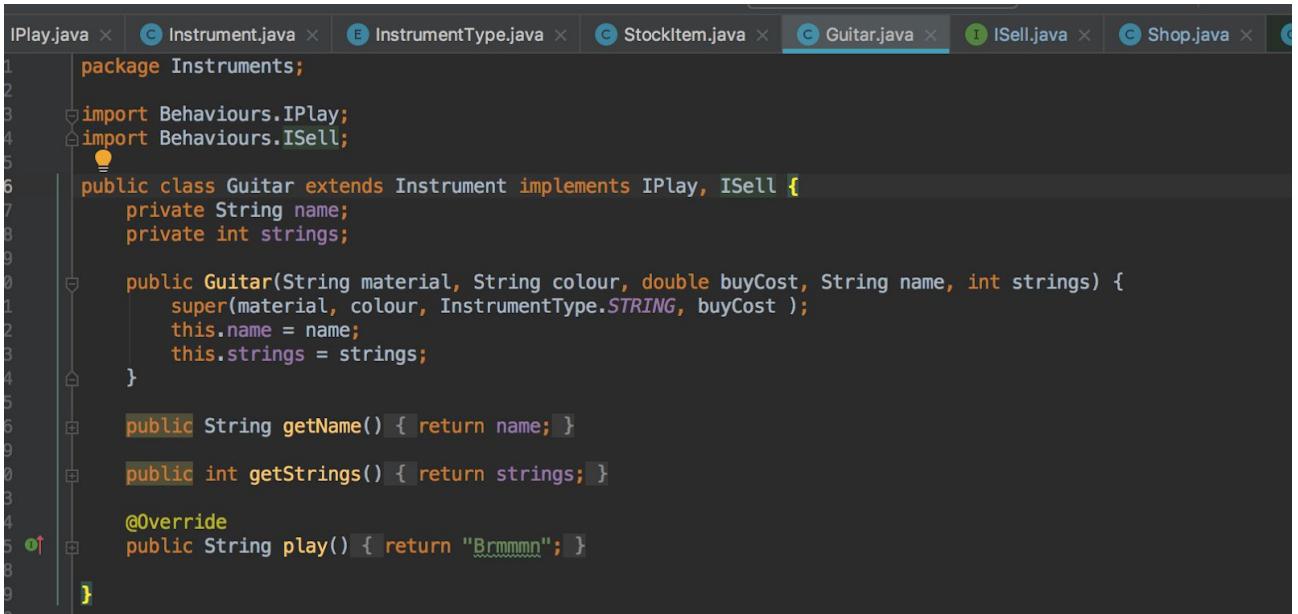
    public String getName() { return name; }

    public double getBuyCost() { return buyCost; }

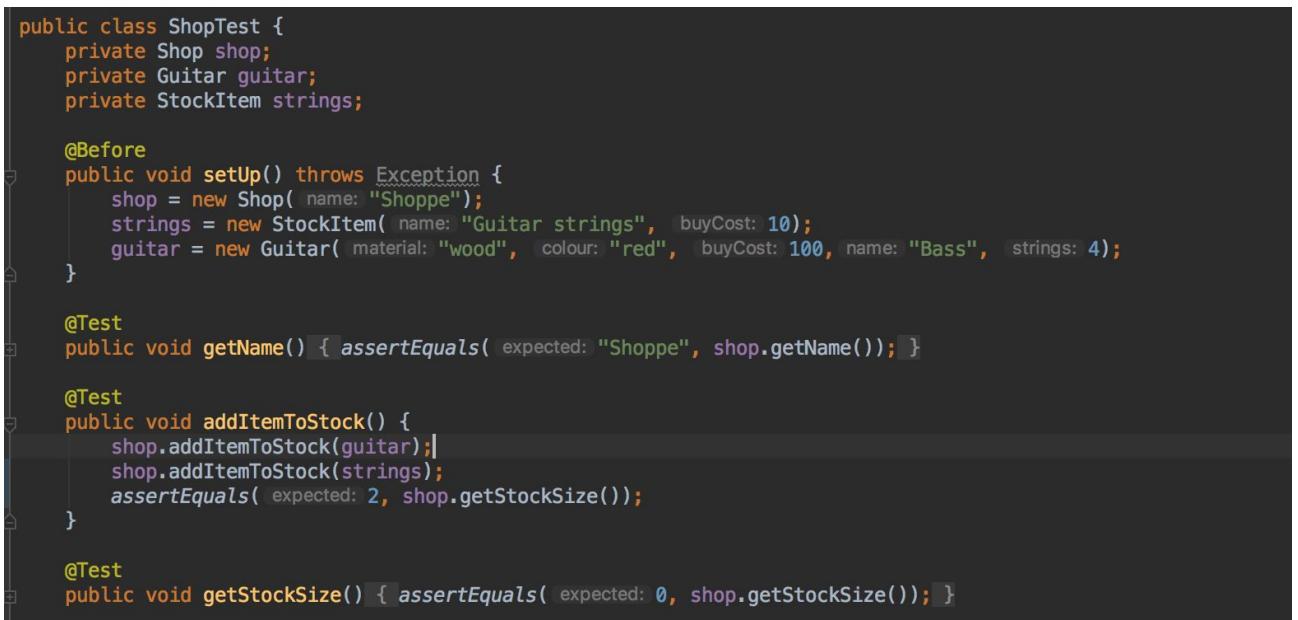
    public double getSellPrice() { return sellPrice; }

    public void setSellPrice(double newPrice) { this.sellPrice = newPrice; }

    @Override
    public double calculateMarkup() {
        return getSellPrice() - getBuyCost();
    }
}
```



```
1 package Instruments;
2
3 import Behaviours.IPlay;
4 import Behaviours.ISell;
5
6 public class Guitar extends Instrument implements IPlay, ISell {
7     private String name;
8     private int strings;
9
10    public Guitar(String material, String colour, double buyCost, String name, int strings) {
11        super(material, colour, InstrumentType.STRING, buyCost );
12        this.name = name;
13        this.strings = strings;
14    }
15
16    public String getName() { return name; }
17
18    public int getStrings() { return strings; }
19
20    @Override
21    public String play() { return "Brmmmn"; }
22}
```



```
public class ShopTest {
    private Shop shop;
    private Guitar guitar;
    private StockItem strings;

    @Before
    public void setUp() throws Exception {
        shop = new Shop( name: "Shoppe");
        strings = new StockItem( name: "Guitar strings", buyCost: 10);
        guitar = new Guitar( material: "wood", colour: "red", buyCost: 100, name: "Bass", strings: 4);
    }

    @Test
    public void getName() { assertEquals( expected: "Shoppe", shop.getName()); }

    @Test
    public void addItemAtStock() {
        shop.addItemToStock(guitar);
        shop.addItemToStock(strings);
        assertEquals( expected: 2, shop.getStockSize());
    }

    @Test
    public void getStockSize() { assertEquals( expected: 0, shop.getStockSize()); }
```

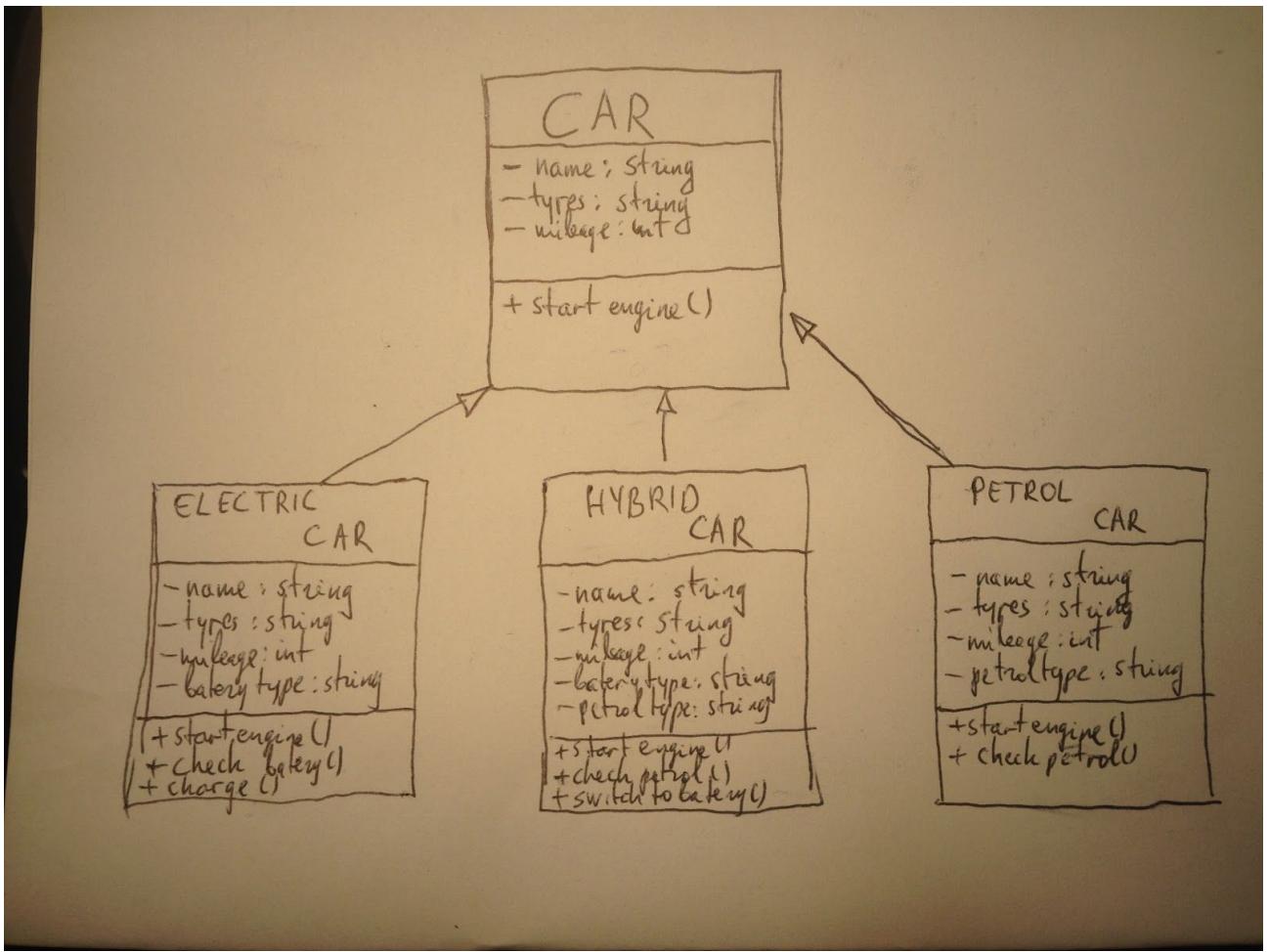
```

4
5     public class Shop {
6         private String name;
7         private ArrayList<ISell>stock;
8
9         public Shop(String name) {
10             this.name = name;
11             this.stock = new ArrayList<>();
12         }
13
14         public String getName() {
15             return name;
16         }
17
18         public void addItemToStock(ISell item){
19             this.stock.add(item);
20         }
21
22         public void removeItemFromStock(ISell item){
23             this.stock.remove(item);
24         }
25
26         public ArrayList<ISell> getStock() {
27             return this.stock;
28         }
29
30         public double gettotalPotentialProfit(){
31             double totalProfit = 0;
32             for(ISell item : stock){
33                 totalProfit += item.calculateMarkup();
34             }
35             return totalProfit;
36         }

```

Description: This program uses polymorphism to be able to treat objects from different classes as having the same ‘form’. By using interface ISell, objects from both Guitar and StockItem are treated as the being same type, when adding to arrayList of stock. When calculating profit it allows to call a method calculateMarkup() on all objects implementing ISell.

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



Description: Electric Car, Hybrid Car and Petrol Car classes inherit name, tyre type, and mileage properties and `startEngine()` method from Abstract Car class. Each of them has distinct variables and methods to deal with their specific needs.

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

```

public abstract class Player {
    private String name;
    protected int hitPoints;
    protected ArmourType armour;
    private WeaponType currentWeapon;
    private ArrayList<WeaponType> weapons;
    protected int gold;
    protected int gems;

    public Player(String name, int hitPoints, ArmourType armour, WeaponType currentWeapon) {
        this.name = name;
        this.hitPoints = hitPoints;
        this.weapons = new ArrayList<>();
        this.armour = armour;
        this.currentWeapon = currentWeapon;
        gold = 0;
        gems = 0;
    }

    public String getName() { return name; }

    public int getHitPoints() { return hitPoints; }
}

```

Description: This Player class has some of its variables set to private and some set to protected. Setting class variable to private encapsulates it, so it's only accessible for this class, and protected variables are accessible outside the class but only can be seen by subclasses or package member.

| 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: | |

```
package Stalls;

import Behaviours.IReviewed;

public abstract class Stall implements IReviewed {
    private String name, ownerName, parkingSpot;
    private int rating;

    public Stall(String name, String ownerName, String parkingSpot) {
        this.name = name;
        this.ownerName = ownerName;
        this.parkingSpot = parkingSpot;
    }

    public String getName() {
        return name;
    }

    public String getOwnerName() {
        return ownerName;
    }

    public String getParkingSpot() {
        return parkingSpot;
    }

    public int getRating() {
        return rating;
    }

    public void setRating(int rating){
        this.rating = rating;
    }
}
```

```

package Stalls;

import Behaviours.ITicketed;
import Visitor.Visitor;

public class IceCreamStall extends Stall implements ITicketed {
    private double defaultPrice;

    public IceCreamStall(String name, String ownerName, String parkingSpot) {
        super(name, ownerName, parkingSpot);
        this.defaultPrice = 2.8;
    }

    @Override
    public double defaultPrice() {
        return this.defaultPrice;
    }

    @Override
    public double priceFor(Visitor visitor) {
        return this.defaultPrice;
    }
}

```

```

public class iceScreamStallTest {
    private IceCreamStall iceCreamStall;

    @Before
    public void before(){
        iceCreamStall = new IceCreamStall( name: "Icycle", ownerName: "Isla", parkingSpot: "23");
    }

    @Test
    public void hasName(){
        assertEquals( expected: "Icycle", iceCreamStall.getName());
    }

    @Test
    public void hasOwnerName(){
        assertEquals( expected: "Isla", iceCreamStall.getOwnerName());
    }
}

```

Description: First screenshot is for abstract class “Stall”, second shows IceScreamStall class which inherits from abstract class. And in the last screenshot we can see object of IceScreamStall class, which can call methods inherited from the abstract class (getName and getOwnerName).

| 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: |

```

public ArrayList<ITicketed> getAllAllowedFor(Visitor visitor){

    ArrayList<ITicketed> allAllowed = new ArrayList<>();
    ArrayList<ISecurity> checks = new ArrayList<>();
    for (Attraction attraction : attractions){
        if(attraction instanceof ISecurity && attraction instanceof ITicketed){
            checks.add((ISecurity) attraction);
        } else if (attraction instanceof ITicketed){
            allAllowed.add((ITicketed) attraction);
        }
    }

    for (Stall stall : stalls){
        if(stall instanceof ISecurity){
            checks.add((ISecurity) stall);
        } else if (stall instanceof ITicketed){
            allAllowed.add((ITicketed) stall);
        }
    }

    for (ISecurity check : checks){
        if(check.isAllowedTo(visitor)){
            allAllowed.add((ITicketed) check);
        }
    }
    return allAllowed;
}

```

Description: This algorithm looks through all the stalls and attractions which require a ticket in a theme park and checks if particular customer is allowed to use them. In the it returns a list of attractions and stalls which pass the requirements. I used this algorithm because it is able to deal with all the polymorphisms and inheritance that is required for this task in an efficient manner.

```

public class Palindrome {

    public static boolean palindrome(String text) {
        String reverse = "";
        int length = text.length();
        int i = 1;
        while (i <= length) {
            reverse += text.charAt(length - i);
            i++;
        }

        if (text.equals(reverse)) {
            return true;
        } else {
            return false;
        }
    }
}

```

Description: This is an algorithm to check if the piece of text is a palindrome (spelled the same backwards and forwards) and returns boolean value. I chose this algorithm because it fastest for this task.