

## PDA Software Development - Level 8

Max Robson

I.T 5 - The use of an Array in a program

```
def initialize(name)
  @name = name
  @stomach = []
end
```

```
def stomach_count
  @stomach.length
end
```

```
p stomach_count
```

```
bear = Bears.new('yogi')
```

```
p bear.stomach_count
```

```
[→ day_02 git:(master) × ruby bears.rb
0
→ day_02 git:(master) ×
```

I.T 6 - The use of a hash in a program

```
users = {
  "Jonathan" => {
    :twitter => "jonnyt",
    :favourite_numbers => [12, 42, 75, 12, 5],
    :home_town => "Stirling",
    :pets => {
      "fluffy" => :cat,
      "fido" => :dog,
      "spike" => :dog
    }
  },
  "Erik" => {
    :twitter => "eriksf",
    :favourite_numbers => [8, 12, 24],
    :home_town => "Linlithgow",
    :pets => {
      "nemo" => :fish,
      "kevin" => :fish,
      "spike" => :dog,
      "rupert" => :parrot
    }
  }
},
```

```
# # 1. Get Jonathan's Twitter handle
p users["Jonathan"][:twitter]
# # 2. Get Erik's hometown
p users['Erik'][:home_town]
# # 3. Get the array of Erik's favourite numbers
```

```
[→ day_03 git:(master) × ruby users.rb
"jonnyt"
"Linlithgow"
[12, 42, 75, 12, 5]
```

### I.T 3 - Searching data in a program

```
const Account = require('./account.js');
const Bank = require('./bank.js');

const bank = new Bank();

const craigsAccount = new Account('Craig', 50, 'Personal');
const ericsAccount = new Account('Eric', 1200, 'Personal');

bank.addAccount(craigsAccount);
bank.addAccount(ericsAccount);

const foundAccount = bank.accountByName('Craig');

console.log(foundAccount);
```

```
Bank.prototype.accountByName = function(name){
  const foundAccount = this.accounts.filter(function(account){
    return account.name === name;
  });
  return foundAccount[0];
}
```

```
[→ bank git:(master) ✕ node app.js
Account { name: 'Craig', value: 50, type: 'Personal' }
→ bank git:(master) ✕ ]
```

### I.T 4 - Sorting data in a program

```
const Dinosaur = require('./dinosaur.js');
const Park = require('./park.js');

const park = new Park();

const barney = new Dinosaur('Purple Thing', 1);
const lily = new Dinosaur('Velociraptor', 3);
const rooney = new Dinosaur('Velociraptor', 2);
const james = new Dinosaur('Pteradactyl', 4);

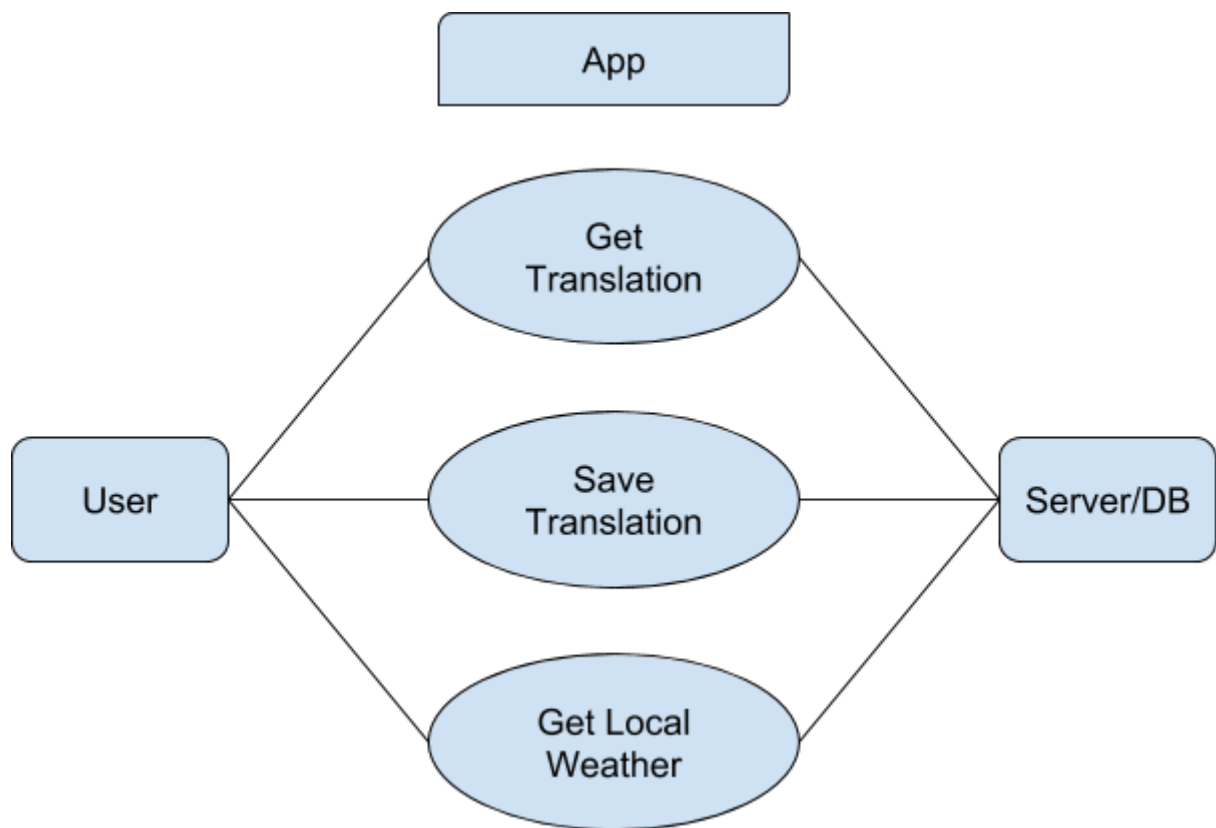
park.addDinosaur(barney);
park.addDinosaur(lily);
park.addDinosaur(rooney);
park.addDinosaur(james);
park.sortByAnnualOffspring();

console.log(park);
```

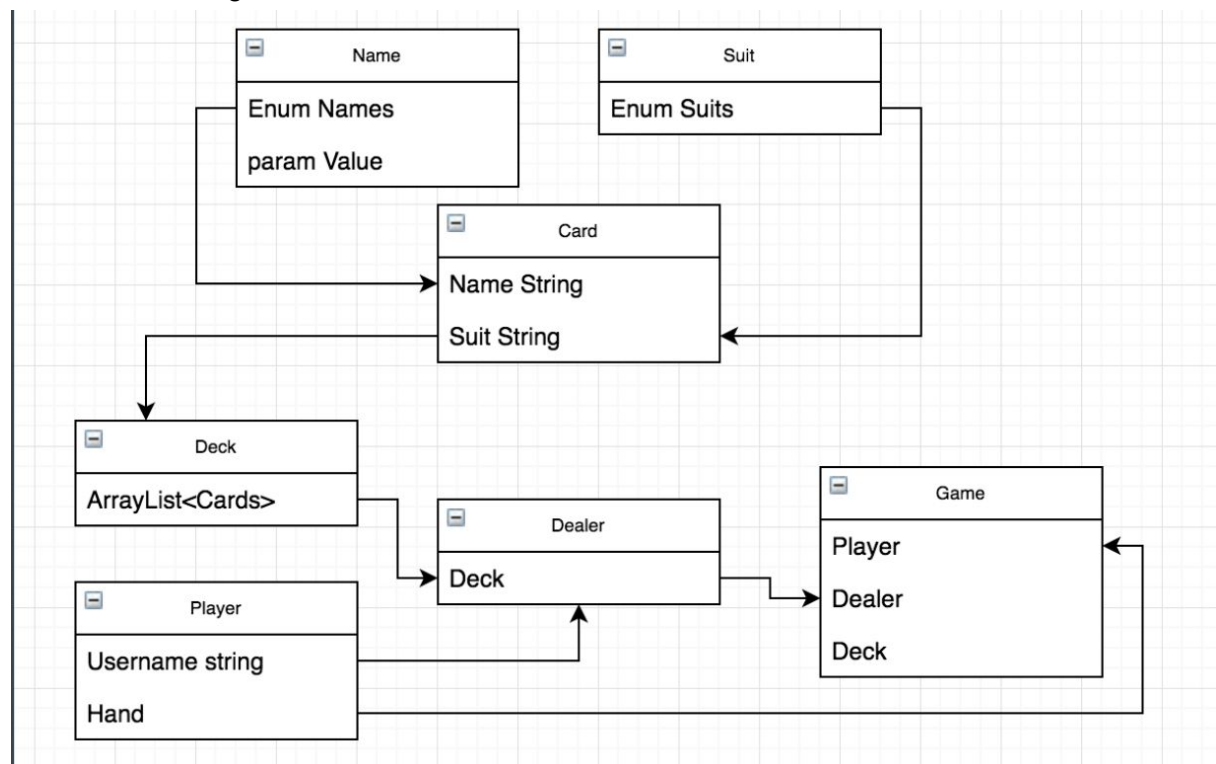
```
Park.prototype.sortByAnnualOffspring = function () {
  this.enclosure.sort(function (a,b){
    return a.annualOffspring - b.annualOffspring;
  });
};
```

```
[→ jurassic_park git:(master) ✕ node app.js
Park {
  enclosure:
    [ Dinosaur { type: 'Purple Thing', annualOffspring: 1 },
      Dinosaur { type: 'Velociraptor', annualOffspring: 2 },
      Dinosaur { type: 'Velociraptor', annualOffspring: 3 },
      Dinosaur { type: 'Pteradactyl', annualOffspring: 4 } ]
}
```

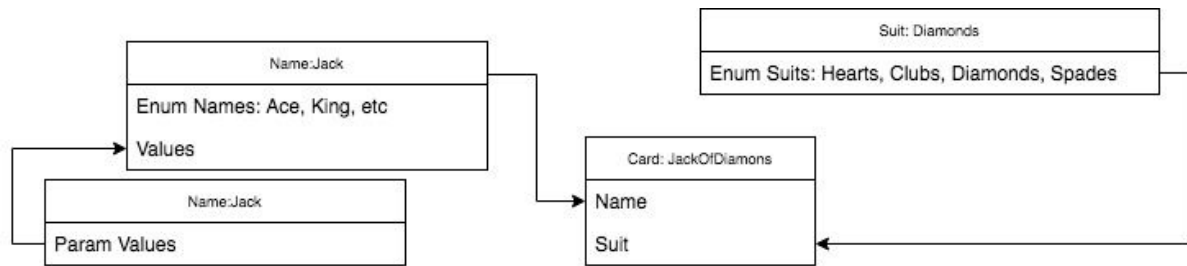
#### A.D 1 - Use case diagram



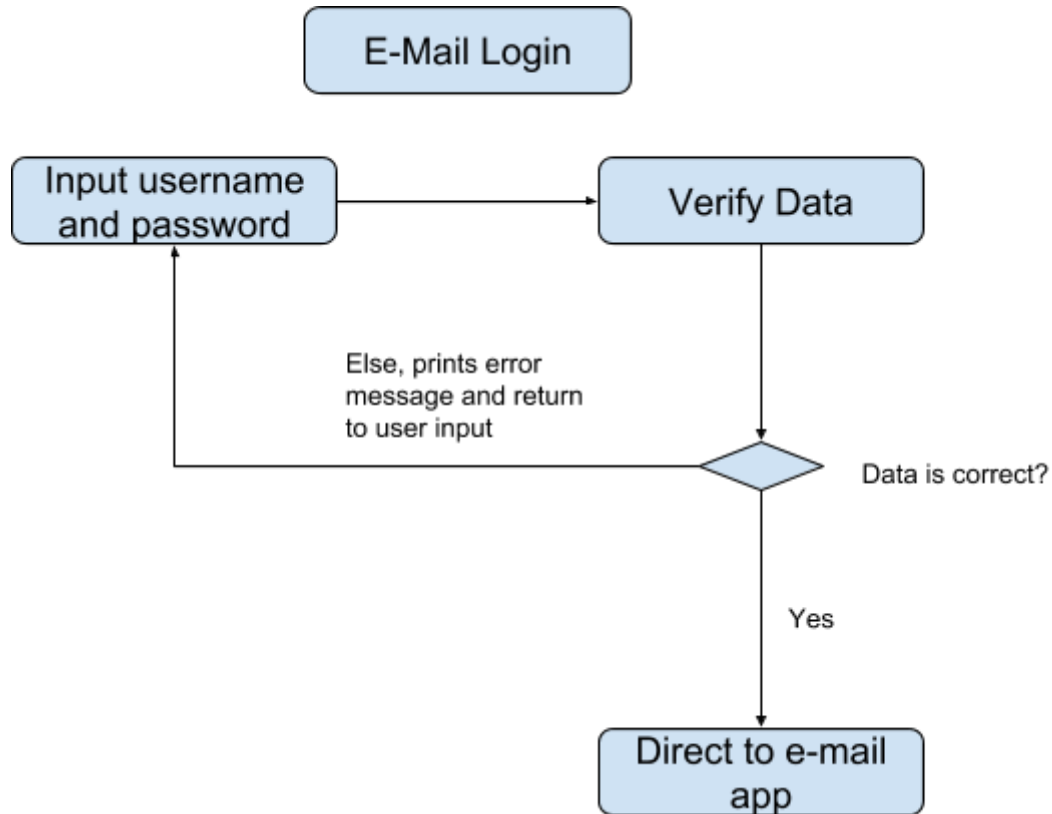
#### A.D 2 - Class diagram



### A.D 3 - Object diagram



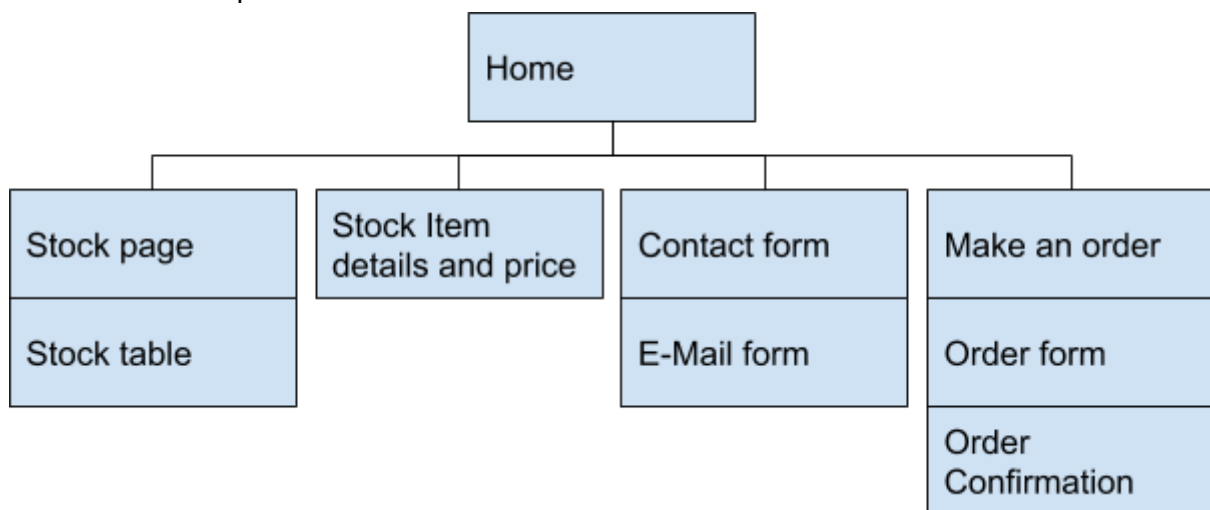
### A.D 4 - Activity diagram



#### A.D 5 - Implementations Constraints diagram

Topic	Constraint	Solution
Software platforms	Not all browsers may support the most recent Java Release	Use a transpiler like babel to convert our program to useable code
Usability	Some users may need screen readers	Be sure to use semantic HTML
Time Limitations	The product owner wants a working alpha in one month	Set project requirements accordingly

#### P 5 - User Sitemap



P 6 - Two wireframe designs

Page 1

https://www.default.com

travelingo

insert name here

submit

Select Home Language

(native language)

(Customised welcome message displays here)

(Flag)

Select Country

(country - language)

insert custom phrase

submit

Phrase	Translation
Hello	Bonjour
Goodbye	Au revoir

Select Additional Info

(additional info displays here)

ROCK, PAPER, SCISSORS!

Computer's move

VS

Player's Move

Result

PLAY

## P 10 - Pseudocode

```
it('should have a favoriteFood', function(){  
  // uses deepStrictEqual to assert that the hero's  
  favorite food is food 1 (Sausage)  
  assert.deepStrictEqual(hero.favoriteFood, food1)  
});
```

## P 13 - Processing user input

Highcharts.com © Natural Earth

Albania - Albanian

Hello, can I have a biscuit? submit

English	Albanian		
	Përshëndetje	speak	delete
I have the bill please ?	Mund ta kam faturën ju lutem?	speak	delete
Where is the bank ?	Ku eshte banka ?	speak	delete
Hello, can I have a biscuit?	Përshëndetje, a mund të kem një biskotë?	speak	delete

## P 14 - Data persistence

Name

Quote

- Some dude - "I like your hat"

```
// CREATE
server.post('/api/quotes', function(req, res){
  const quotesCollection = db.collection('quotes');
  const quoteToSave = req.body;
  console.log(quoteToSave);
  quotesCollection.save(quoteToSave, { log(number) : number, result: true }, function(err, result){
    if (err){
      console.log(err);
      res.status(500);
      res.send();
    }

    res.status(201);
    res.json(result.ops[0]);

    console.log('Saved to db');
  });
});
```

```
promiseLibrary: [Function: Promise],
readConcern: undefined } }
[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
Listening on port 3000
Connected to database
{ name: 'Some dude', quote: 'I like your hat' }
Saved to db
□
```

## P 15 - Correct output of results and feedback

*Already shown above???*

## I.T 7 - Polymorphism

```
1 public interface IConnect {
2     String connect(String data);
3 }
4
```

```
public class InternetRadio implements IConnect {
    private String name;

    public InternetRadio(String name) { this.name = name; }

    public String connect(String networkName) { return "In"; }
```

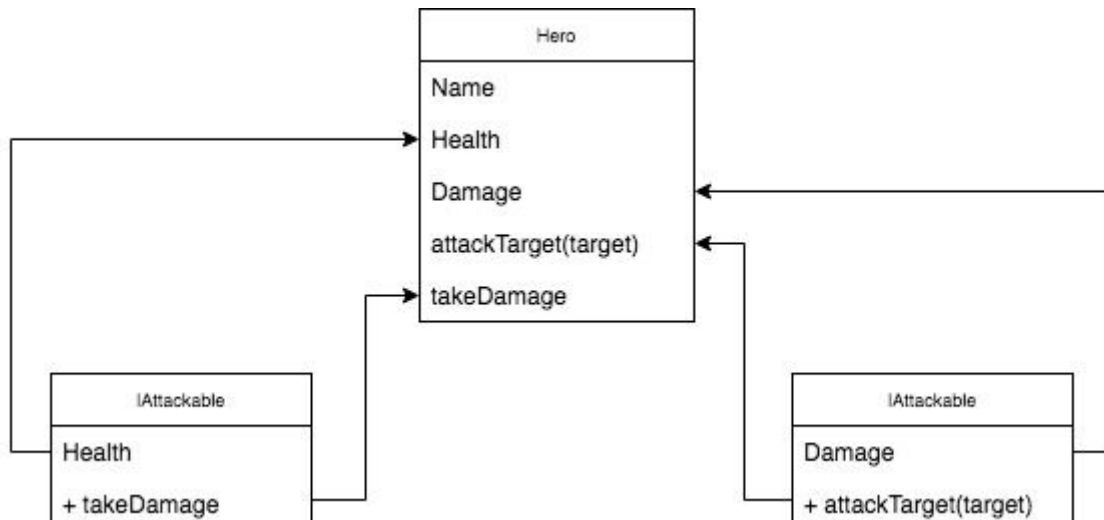
```
public class Desktop implements IConnect {
    private String name;
    private String make;
    private String model;

    public Desktop(String name, String make, String model) {
        this.name = name;
        this.make = make;
        this.model = model;
    }

    public String connect(String networkName) { return "In"; }
```



## A.D 5 - Inheritance Diagram



## I.T 1 - Encapsulation

```
public abstract class Instrument implements ISellable, IPlayable {

    private int retail;
    private int wholesale;
    private String colour;
    public String type;

    public Instrument(String colour, String type, int retail, int wholesale){
        this.colour = colour;
        this.type = type;
        this.retail = retail;
        this.wholesale = wholesale;
    }

    public abstract String play();

    public int getRetail() { return this.retail; }

    public int getWholesale() { return this.wholesale; }

    public int markup() { return retail - wholesale; }

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

## I.T 2 - Inheritance in a program

```
public abstract class Instrument implements Comparable<Instrument> {  
  
    private int retail;  
    private int wholesale;  
    private String colour;  
    public String type;  
  
    public Instrument(String colour, int retail, int wholesale, String type) {  
        this.colour = colour;  
        this.type = type;  
        this.retail = retail;  
        this.wholesale = wholesale;  
    }  
  
    public abstract String play();  
}
```

```
public class PercussionInst extends Instrument {  
  
    public String subType;  
  
    public PercussionInst(String type, String colour, String subType, int retail, int wholesale) {  
        super(colour, type, retail, wholesale);  
        this.subType = subType;  
    }  
  
    public String play() { return "Playing " + this.subType; }  
}
```

## P 11 - GitHub link working alone

added some styling to the selector and text input

Browse files

🔗 master

 Splodge77 committed 12 days ago

1 parent [fc6dd86](#) commit [eb3328c77b22843c3c5a6ff1a1f357c65dca3509](#)

Showing 3 changed files with 25 additions and 5 deletions.

Unified Split

20  client/build/css/phrases.css

View

@@ -5,8 +5,13 @@

5 margin-left: 5%;

6 }

7

8 -table{

9 - /\* border: 1px solid black; \*/

8 +table, td, th{

9 + border: 1px solid grey;

10 +}

11 +

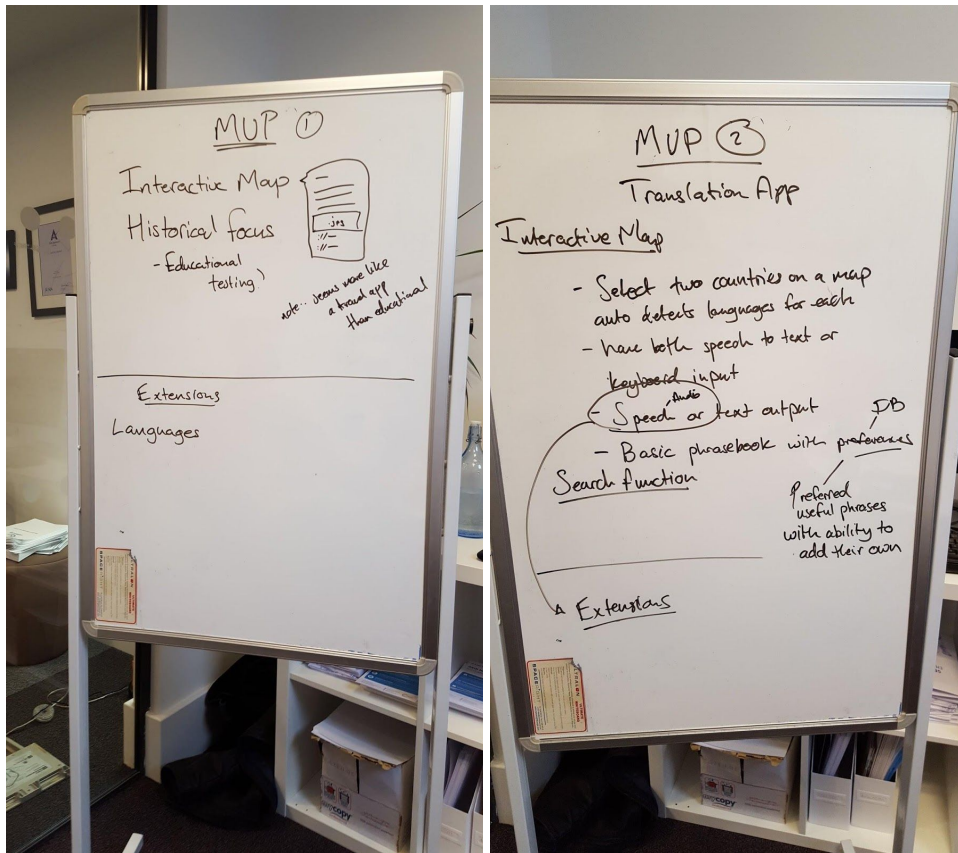
12 +select{

13 + background-color: rgb(149, 228, 249);

14 + height: 26px;

10 15 }

P 12 - Photos of planning and changes in development



## P 18 - Use of testing in program

```
const assert = require('assert');
const Orc = require('../models/npcs/orc.js');
const Weapon = require('../models/environment/weapon.js');
const Hero = require('../models/characters/hero.js')

describe("Orc Test", function(){

  let orc;
  let weapon;
  let hero;

  beforeEach(function(){
    orc1 = new Orc("Lok-tar", weapon1)
    weapon1 = new Weapon("Axe", 20, 2)
    hero1 = new Hero("Sir.JsWat", "Bacon", weapon1)
  });

  it('should have a name', function(){
    assert.deepStrictEqual(orc1.name, "Lok-tar")
  });

  it('should have a base health', function(){
    assert.deepStrictEqual(orc1.health, 150)
  });
});
```

47 passing (40ms)

1 failing

1) Orc Test  
should be armed:

AssertionError [ERR\_ASSERTION]: 50 deepStrictEqual 70  
+ expected - actual

-50  
+70

at Context.<anonymous> (client/src/specs/orc\_spec.js:16:12)

```
it('should be armed', function(){
  orc1.isArmed();
  assert.deepStrictEqual(orc1.damage, 50)
});
```

### Orc Test

- ✓ should have a name
- ✓ should have a base health
- ✓ should have a base damage
- ✓ should start with a weapon
- ✓ should be armed
- ✓ should be disarmed
- ✓ can attack a hero
- ✓ should be dead
- ✓ should not attack if dead

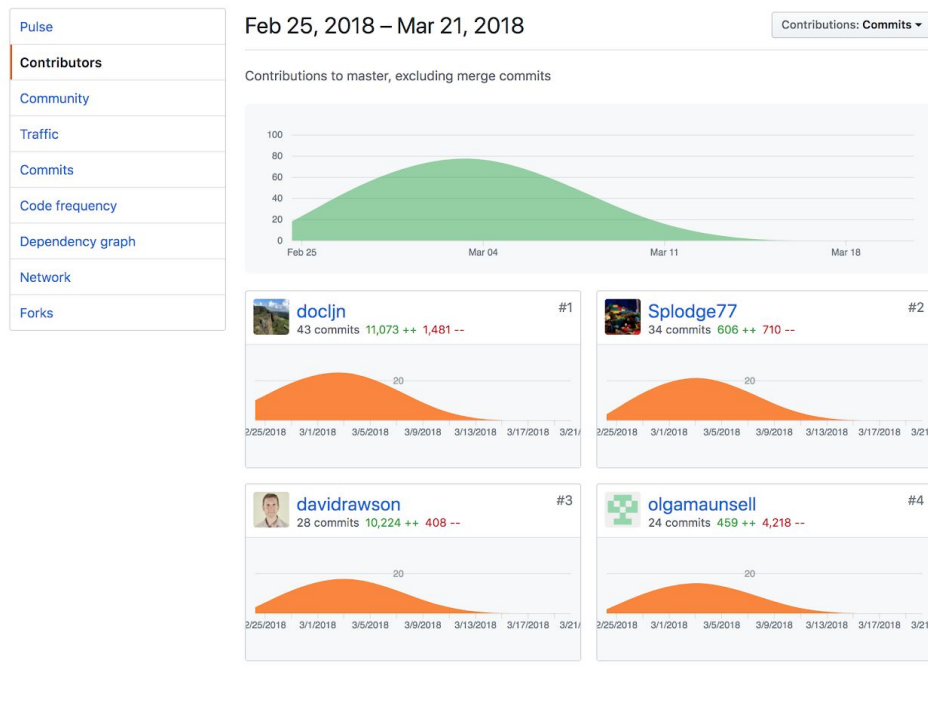
## P 16 - Use of an API

```
const app = function () {  
  const url =  
    "https://s3-eu-west-1.amazonaws.com/brewdogapi/beers.json";  
  
  const select = document.querySelector('#beer-select');  
  
  makeRequest(url, requestComplete);  
};
```

```
const populateList = function(beers){  
  const dl = document.getElementById("beer-list");  
  beers.forEach(function(beer){  
    const dt = document.createElement('dl');  
    dt.innerText = beer.name;  
    const image = document.createElement('img')  
    image.src = beer.image_url;  
    image.alt = beer.name;  
    image.width = '80';  
    image.height = '240';  
    dl.appendChild(dt);  
    dl.appendChild(image);  
  });  
};
```



## P 1 - GitHub contributors page for group project



## P 2 - Project brief from group project

### Multi-Language Travel Phrase Web App - High Level Design

#### MVP

##### Welcome page

- on 1st visit of app allows user to enter their name and home country

##### Translation View

- Provide a drop down list of destination countries.
- On country selection displays the country's default language and top 10 travel phrases.
- Provide an option to save favourite country/language pairing & phrases for later use.

#### Extensions

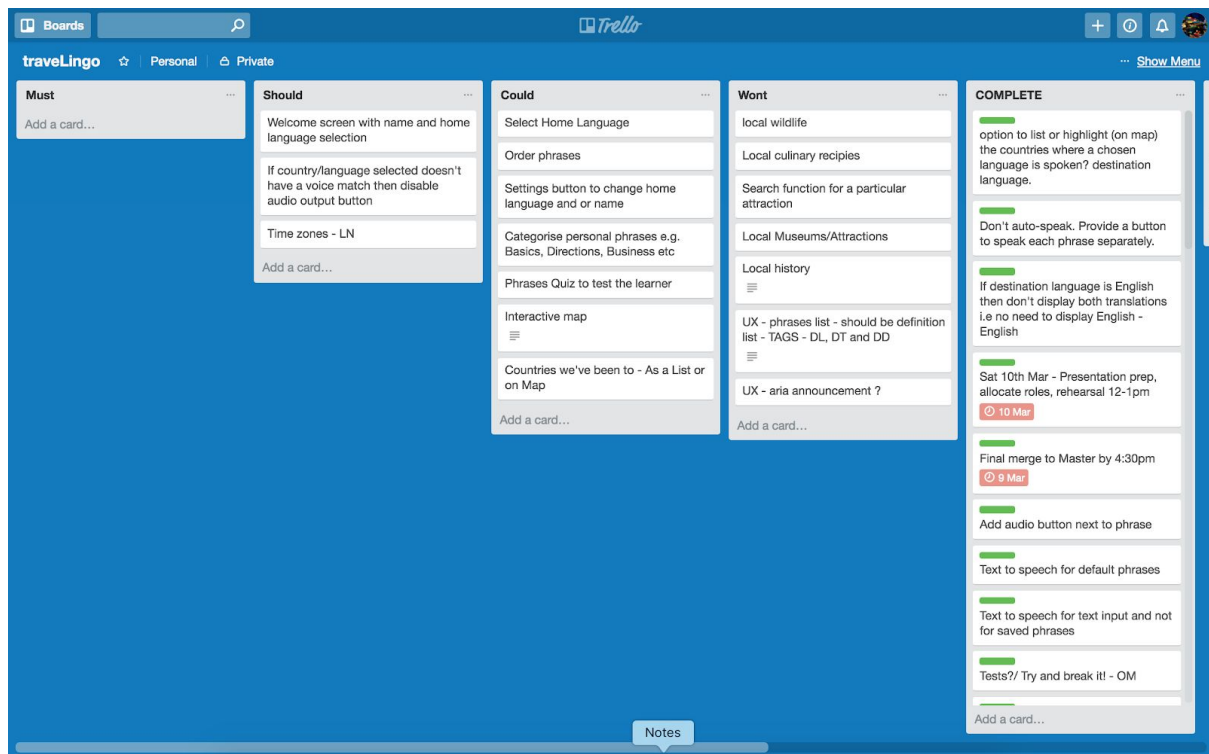
##### Priority

- On subsequent visits to web app displays welcome greeting in home language e.g. "Bonjour Jean-Marie"
- Create a settings option to allow user to change name and home country
- User can enter their own phrase to translate and save if desired
- Map view showing visited countries / planned journeys
  - this could also be shown in the text version with a simple flag

##### Consider

- User can set the display order of phrases
- Map interaction to show country information
- Text to speech output for pronunciation
- Local weather
- Local timezones

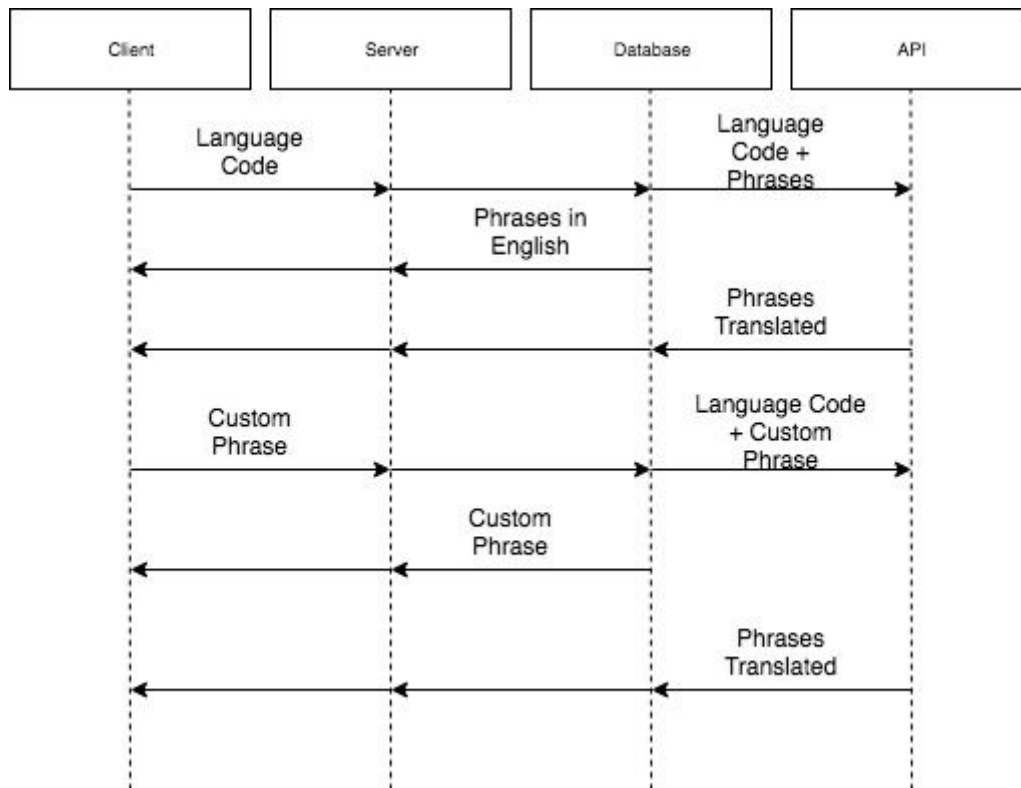
### P 3 - Planning for group project



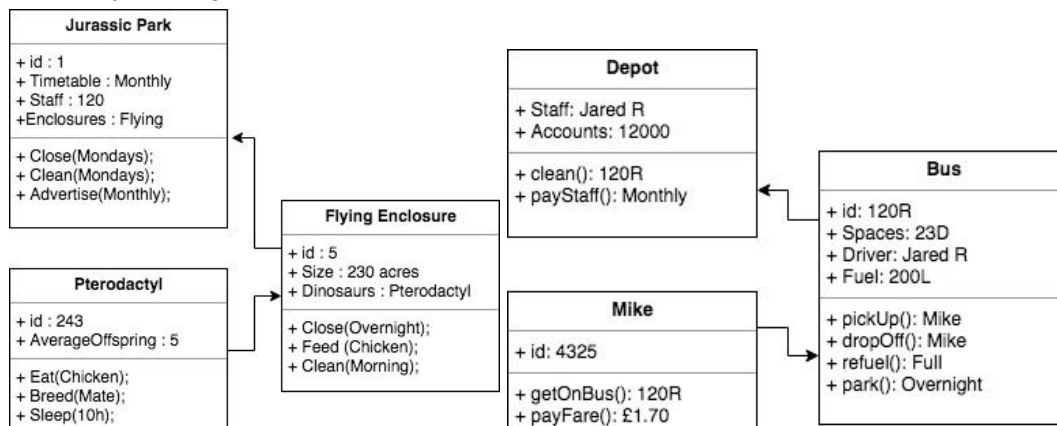
### P 4 - Acceptance Criteria and Test Plan

Acceptance Criteria	Expected Output/Result	Pass/Fail
User should be able to access a list of common phrases to be translated	Common phrases in English and selected language	Pass
User should see local weather	Local weather for selected country's capital city is displayed	Pass
User should see local attractions	Local attractions displayed in separate div when country selected	Fail

## P 7 - System interaction diagrams



## P 8 - Object diagrams





## P 9 - Algorithms

```
public String checkWinner(){
    for (Player player : this.players){
        if (player.getScore() > dealer.getScore() && player.playerIsBust()==false){
            return "Player Wins";
        }
    } return "Dealer Wins";
}
```

```
public int setScore(){
    for(Card card : hand){
        int value = card.getCardValue();
        initialScore += value;
        if(value == 11 && initialScore > 21){
            score += 1;
        }
        else score += value;
    }
    return score;
}
```

## P 17 - Bug Tracking Report

User must be able to save custom phrases	Failed	Phrase saved to database instead of local storage	Passed
Translations language must be displayed	Failed	Unsupported Languages filtered out	Passed
Phrase input must be hidden when app starts	Failed	Input cannot have flex and hidden properties. Flex removed.	Passed
Correct language must be displayed on select	Failed	Changed the target.index to -1 so that it misses out the first placeholder in array	Passed
Original phrase and translation must appear different	Failed	HTML element id's assigned in JavaScript so that the can be targeted by CSS	Passed