

# Evidence for Project Unit

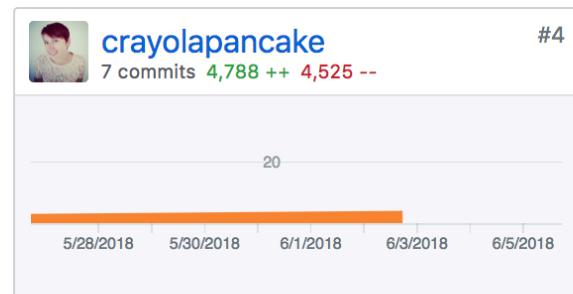
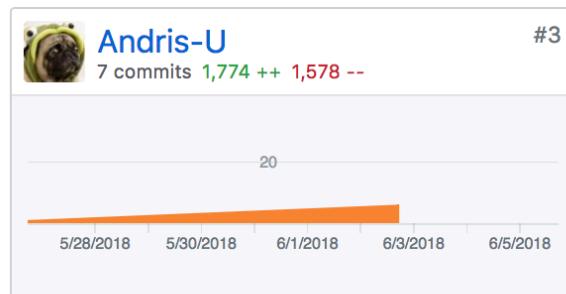
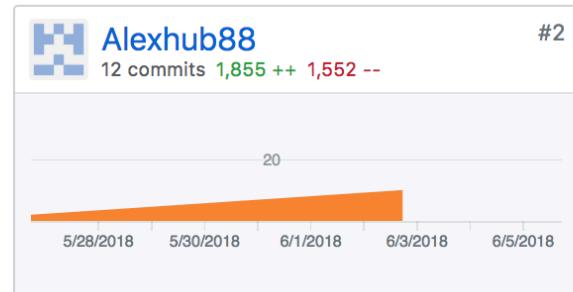
Jaime Lopez  
Cohort E20

## P. 1 Github Contributors page

May 27, 2018 – Jun 6, 2018

Contributions: Commits ▾

Contributions to master, excluding merge commits



## P. 2 Project Brief

### Pub Quiz

---

A chain of pubs is looking to improve its system of quizzes to adapt themselves to modern times. They have decided to create an interactive webApp to display questions, answers and stats of the team in real time. Your task is to make an MVP so they can play the quiz with different number of players and being able to configure number of questions, categories and difficulty. They want questions to be random and numerous, so you might use an API to bring in content to the app.

### MVP

---

- There is a screen to set up the game:
  - Number of players
  - Category of the questions
  - Difficulty of the questions
  - Number of questions
- There are players
- There are questions
- Questions have multiple answers
- Player select answers taking turns
- Correct answer will be displayed after all the players answer the question
- After all the questions, game shows the winner

### Extensions

---

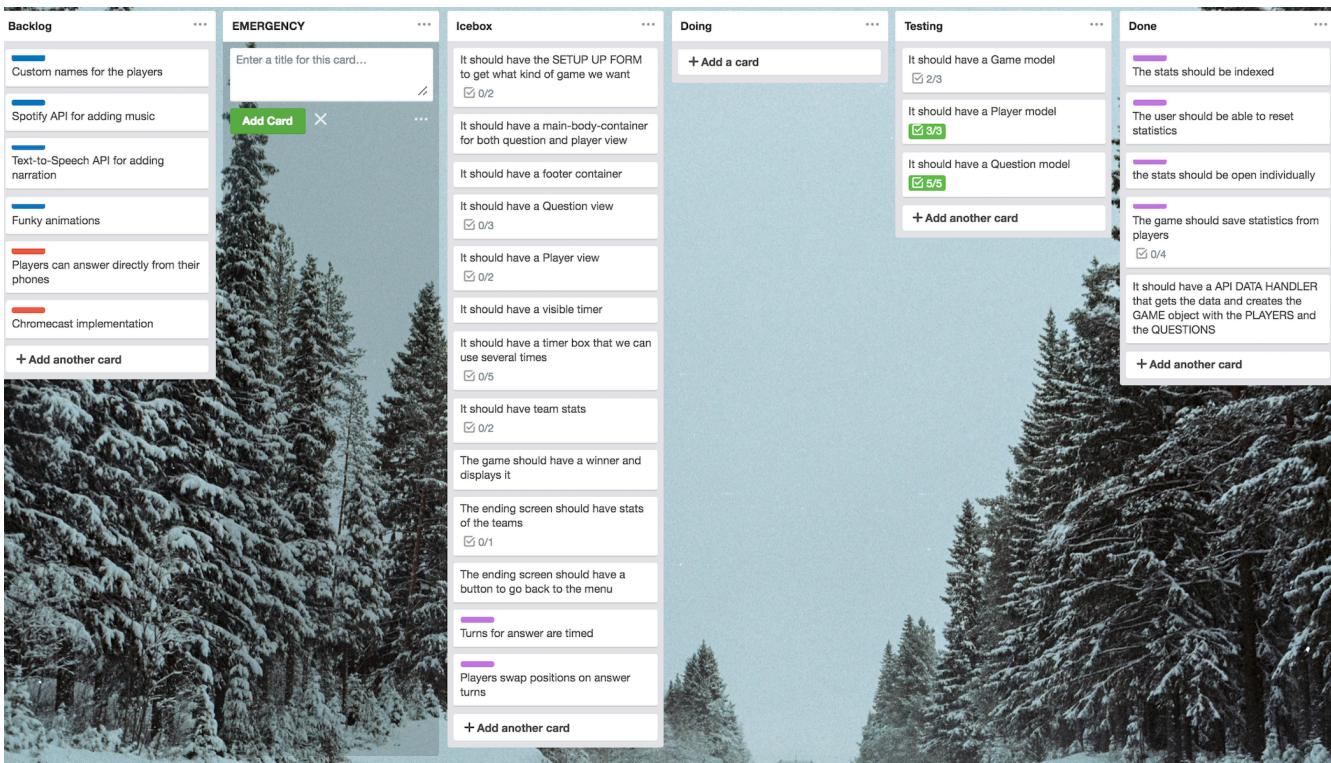
- There are stats for every player
- Stats can be recorded on a high score chart
- The turns of the players are randomly decided
- Turns are timed. Every consequent player will have less time.

### API, Libraries, Resources

---

- <https://www.highcharts.com/> HighCharts is an open-source library for rendering responsive charts.
- <https://opentdb.com/> OpenTriviaDB is an open-source library of trivia questions.

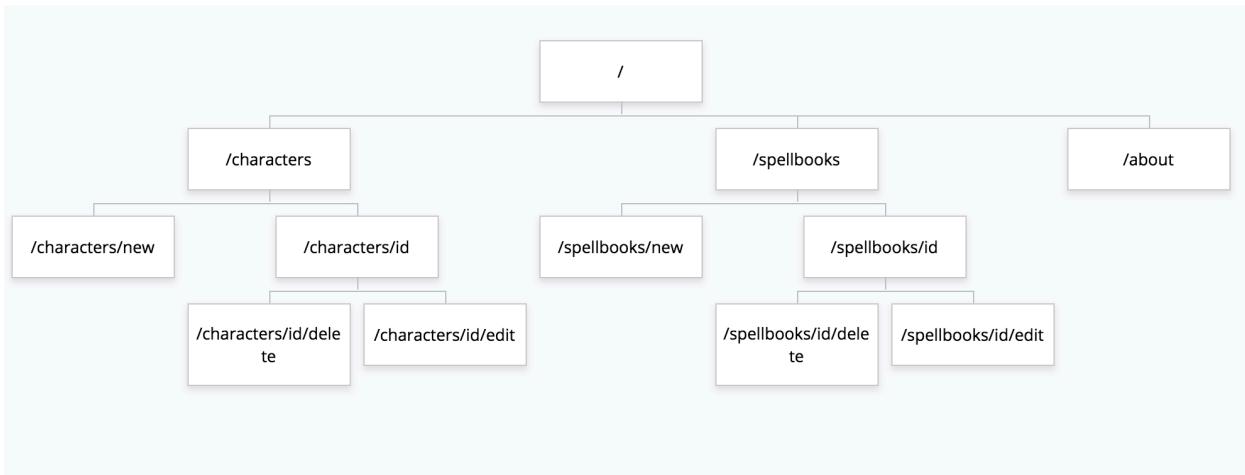
## P. 3 Use of Trello



## P. 4 Acceptance Criteria / Test plan

Acceptance Criteria	Expected Result / Output	Pass / Fail
Player can set number of player	Game creates a player object for each player and show them on the game screen	Pass
Player can set number of questions	Game takes from API a number of questions equal to the value specified by the user	Pass
Player can set difficulty of the questions	Only questions from the specified difficulties will be taken from the API	Pass
Player can set category of the questions	Only questions from the category selected will appear	Pass
Player can answer the question by clicking the answer	Answer selected change color when hovering and add the answer to the player object	Pass
After last player answer, game should show correct answer and pass to the next question	Correct answer is displayed in green and changes question after one second delay	Pass
After last question, game should show end screen	After one second delay, end scene appears showing stats and the winner	Pass

## P. 5 User sitemap



## P. 6 Wireframes designs

The wireframe shows a user interface for managing characters. At the top, there is a header bar with navigation icons (back, forward, search) and links for Dashboard, List all Characters, Spell List, Spellbook list, and About. A logo icon is also present.

The main content area displays "Latest updated characters" with four placeholder cards, each containing a large square with an 'X' and a heading "Character". Below each card is a block of placeholder text and three buttons: "Spellbooks", "Review Spellbook", and "Add Spellbook".

To the right, there is a sidebar with a heading "Heading 2" and more placeholder text. It also contains three buttons: "Spellbooks", "Review Spellbook", and "Add Spellbook".

At the bottom right, there are three small square icons with an 'X' symbol.

ooo

← → ⌂

[Dashboard](#) | [List all Characters](#) | [Spell List](#) | [Spellbook list](#) | [About](#)

Logo

**Spell name**  
School of magic and level

**Description**

*Placeholder text for spell description*

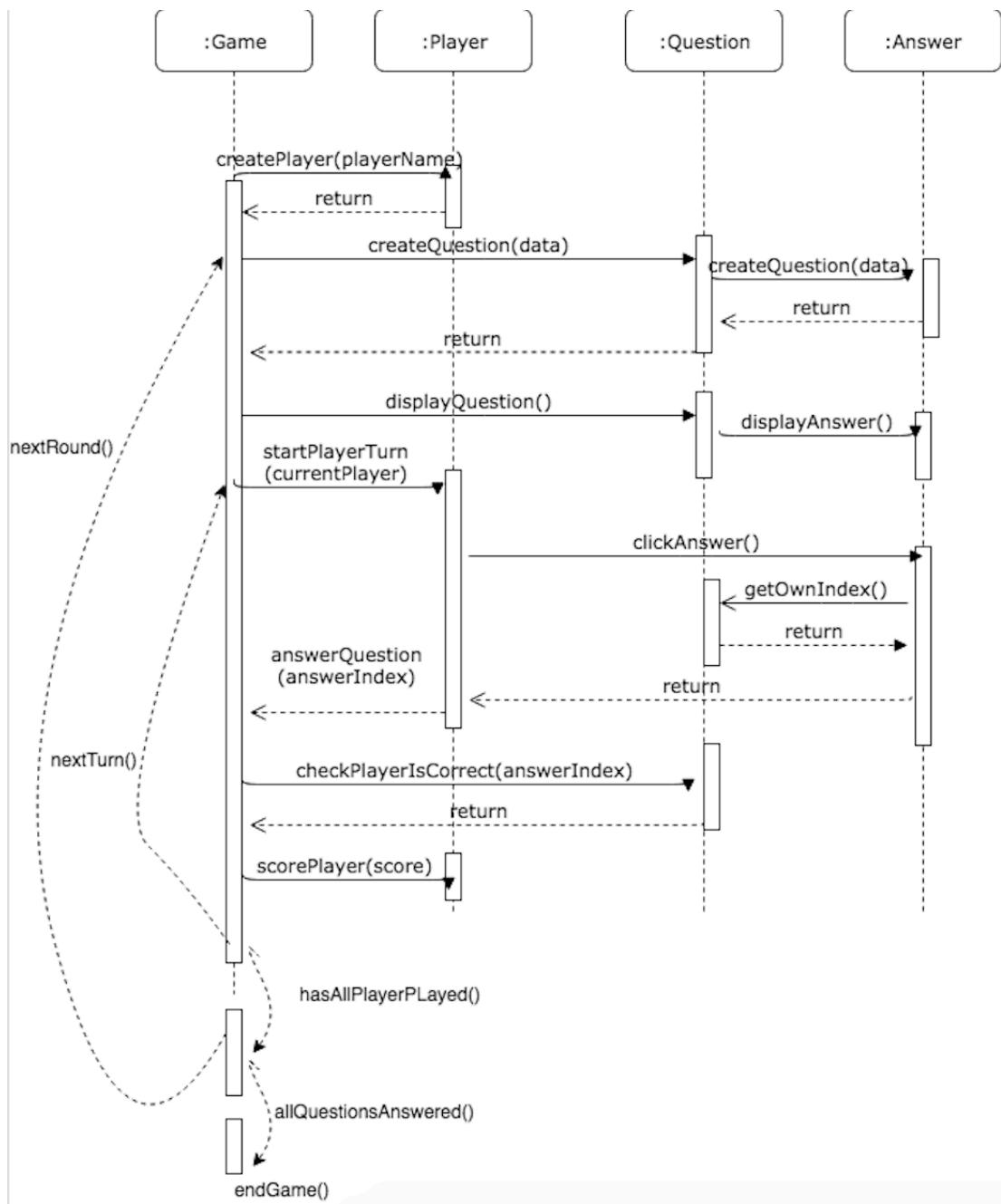
**Add spell to character**

**Select the spellbook**

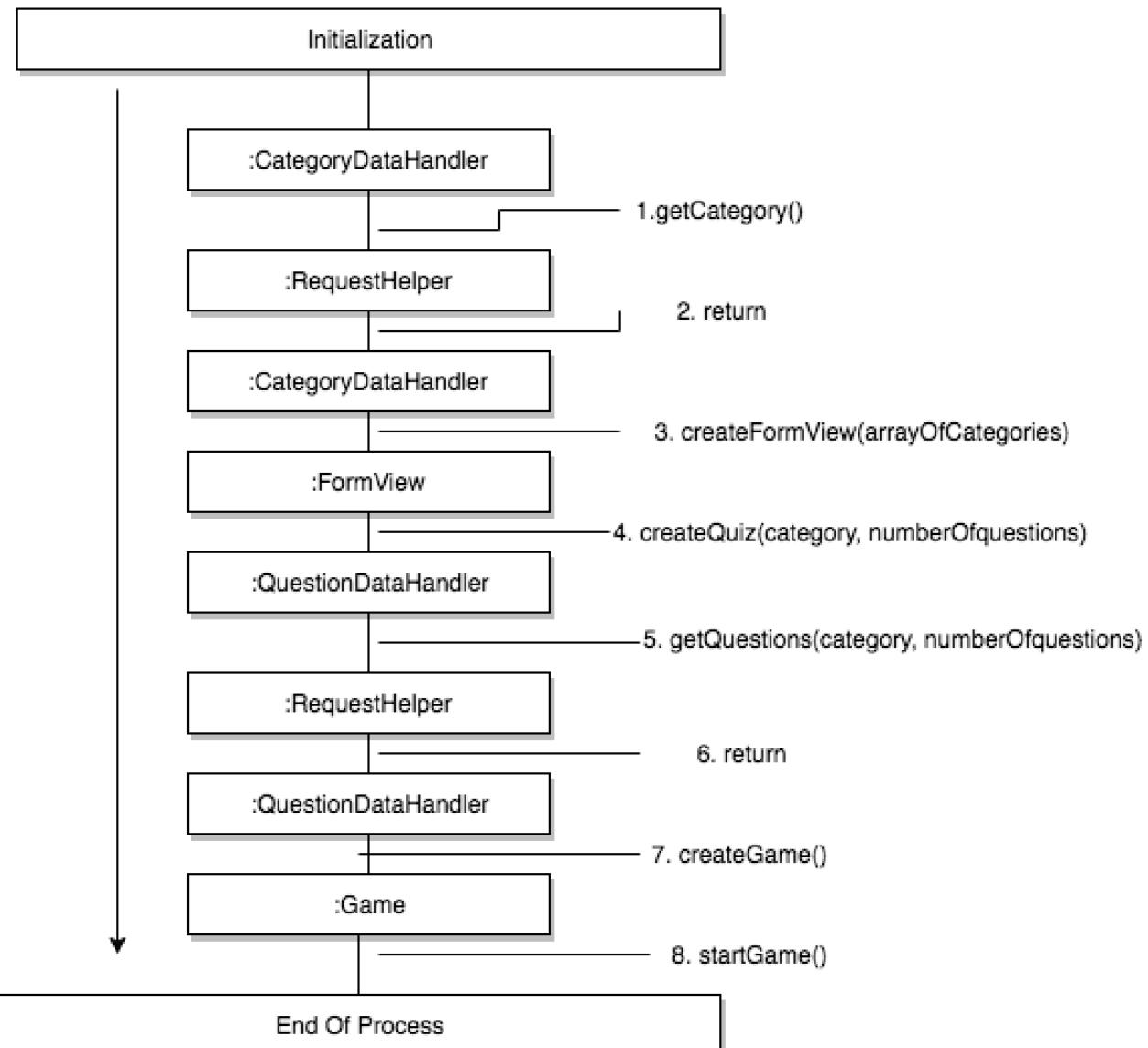


*Placeholder text for spell image or preview*

## P. 7 System interactions diagrams



Collaboration Diagram: Game Start



## P. 8 Two Object Diagrams

Player
+ name: String + score: Integer + correctQuestionsAnswered: Array<Question> + questionsAnswered: Array<Question>

Jaime: Player
+ name: Jaime + score: 50 + correctQuestionsAnswered: [Question1, Question2] + questionsAnswered: [Question1, Question2, Question3, Question4]

Alex: Player
+ name: Alex + score: 25 + correctQuestionsAnswered: [Question2] + questionsAnswered: [Question1, Question2, Question3, Question4]

Question
+ questionText: String + category: String + difficulty: String + correctAnswer: String + allAnswers: Array<String>

question1: Question
+ questionText: "In what year did Neil Armstrong and Buzz Aldrin land on the moon?" + category: "History" + difficulty: "Medium" + correctAnswer: "1969" + allAnswers: ["1965", "1963", "1969", "1975"]

question1: Question
+ questionText: "In what year did Neil Armstrong and Buzz Aldrin land on the moon?" + category: "History" + difficulty: "Medium" + correctAnswer: "1969" + allAnswers: ["1965", "1963", "1969", "1975"]

**P. 9 Choice of two algorithms (find the algorithms on a program you might have written, show the code you have used. )**

```
const PangramFinder = function (phrase) {
  this.alphabet = 'qwertyuiopasdfghjklzxcvbnm'.split('');
  this.phrase = phrase.toLowerCase();
}

PangramFinder.prototype.isPangram = function () {
  return this.alphabet.every( (letter) => {
    return this.phrase.includes(letter);
  });
}

module.exports = PangramFinder;
```

**What it does?** This algorithm checks if an specific phrase is a Pangram (a sentence using every letter of a given alphabet at least once).

**How it does?** Iterate over this.alphabet (an array containing all the letters of the alphabet) and check (returning true or false) if every item of this array applies the condition inside it. The conditions checked inside ‘every()’ return true if the given phrase includes the current letter on the iteration.

So, given the phrase “the quick brown fox jumps over the lazy dog”, it will iterate over the alphabet and check if all the letters inside the array are contained on the phrase. If at least one of them is false, will return false (it is not a pangram, then).

```

//This method is for looking for an object on an array by passing a string containing a verb and a name
private IFoundable findByName(String stringToLook, ArrayList<? extends IFoundable> arrayToLook) {
    //Split the string to look in individual words
    String[] stringArray = stringToLook.split(" ");
    IFoundable foundObject = null;

    //Loop over the words to check if any of the words in the name is included on the array with the words
    //from the action
    firstLoop: for (IFoundable foundable: arrayToLook) {
        for (String foundableWord: foundable.getName().split(" ")) {
            if (Arrays.asList(stringArray).contains(foundableWord)) {
                foundObject = foundable;
                break firstLoop;
            }
        }
    }
    return foundObject;
}

```

**What it does?** This algorithm checks if an specific object exists on a list of objects and returns it only if it exists.

**How it does?** The algorithm takes a string (the user input. For example: ‘attack dragon’) and an array of objects that can be found (for example, [donkey, guard, sword, dragon]). It splits the user input on a array of words [attack, dragon], then, after allocating an empty spot for storing the object (IFoundable foundObject = null), start iterating with a for loop on the array of objects.

Then, the name of every object is split in parts (‘red dragon’ now is [‘red’, ‘dragon’] and iterates with a for loop over this new array of strings. Inside this, if any of the words of the object name match any of the words inputted by the user, save the object on the variable allocated at the beginning and breaks the whole loop.

After this, return the object matching user input, or returns null if no objects match the criteria.

## P. 10 Example of Pseudocode

```
def self.filter_by_school(value)
  #This method receives the input of the user
  # The input is a string selected from a dropdown menu offering
  # all the schools options (an array)
  # Then, with this, this method selects all the entries on a
  # SQL database where the School column shows the value
  # selected by the user
  # The method uses this entries and create an array
  # Populates the array with objects using the entries found in the
  # db.
  # Returns the array of objects
end
```

## P. 11 Github link to one of your projects

<https://github.com/DetectiveAzul/compendium>

The screenshot shows the GitHub repository page for 'compendium' owned by 'DetectiveAzul'. The page includes a header with repository statistics (0 stars, 0 forks), navigation links (Code, Issues, Pull requests, Projects, Wiki, Insights, Settings), and a main content area displaying commit history and file changes.

A DnD spell compendium with capacity to keep track of your characters spells knowledge and spellbooks

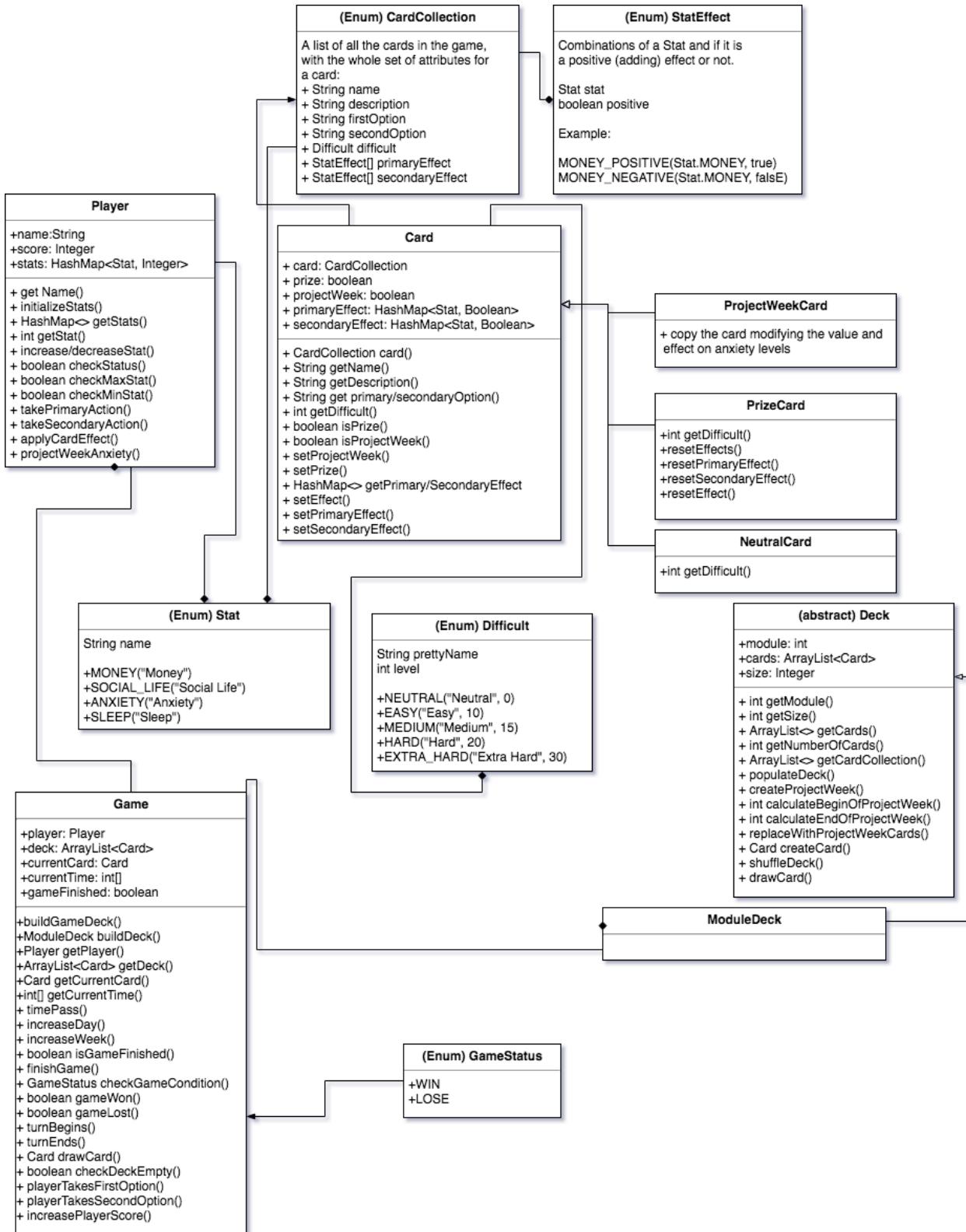
Add topics

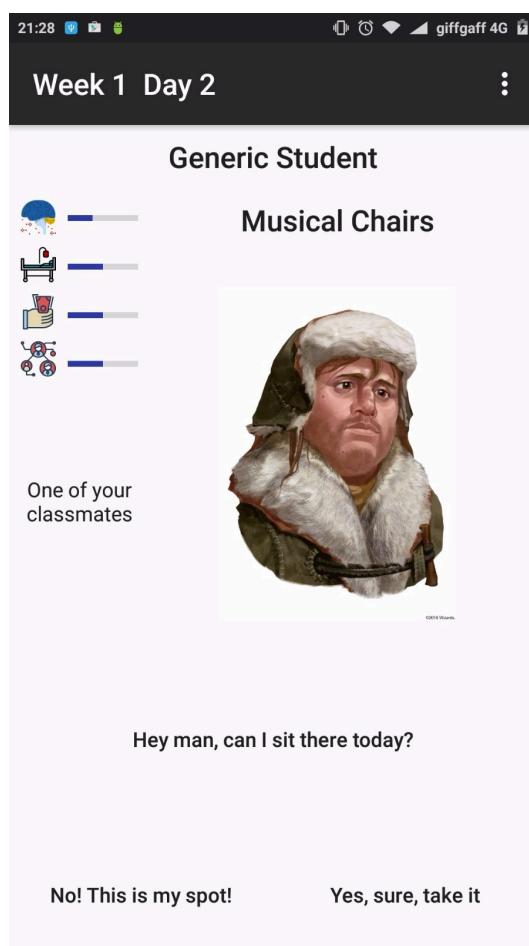
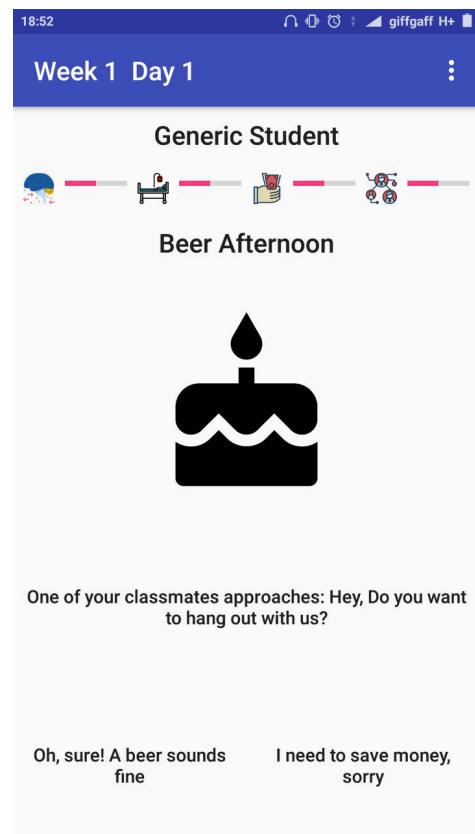
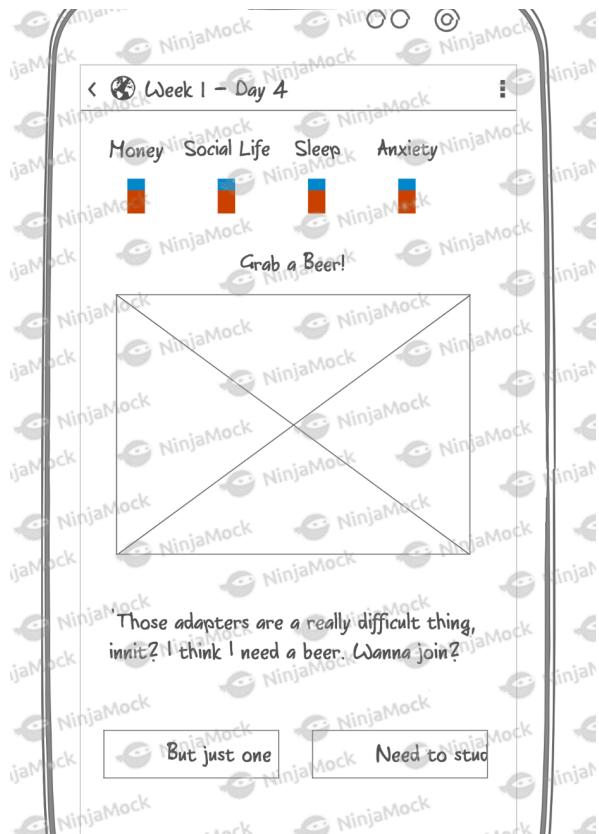
103 commits | 1 branch | 0 releases | 1 contributor

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

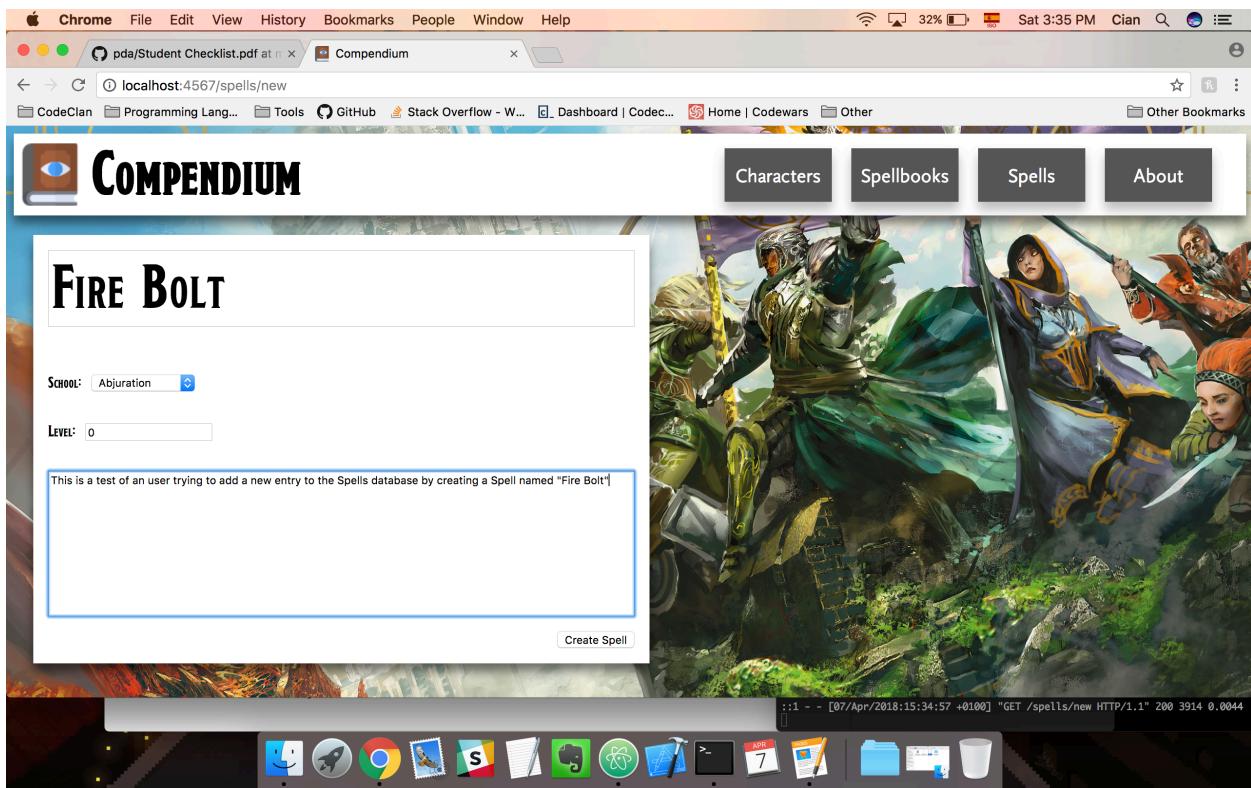
File	Description	Time Ago
DetectiveAzul Showing breaks on spell show.erb	Latest commit 7fb27d7 14 days ago	
controllers	Added 404 website	22 days ago
db	Fixed sql	14 days ago
models	fixed error on method to cheack if spell is repeated on a book	23 days ago
public	Showing breaks on spell show.erb	14 days ago
views	Fixed tables bug when inputing very long names	22 days ago
Gemfile	README and gem files	23 days ago
Gemfile.lock	README and gem files	23 days ago
Procfile	Heroku deployment	14 days ago
README.md	README and gem files	23 days ago
app.rb	Heroku deployment	14 days ago
config.ru	Heroku deployment	14 days ago

## P. 12 Screenshot of your planning and the different stages of development to show changes.

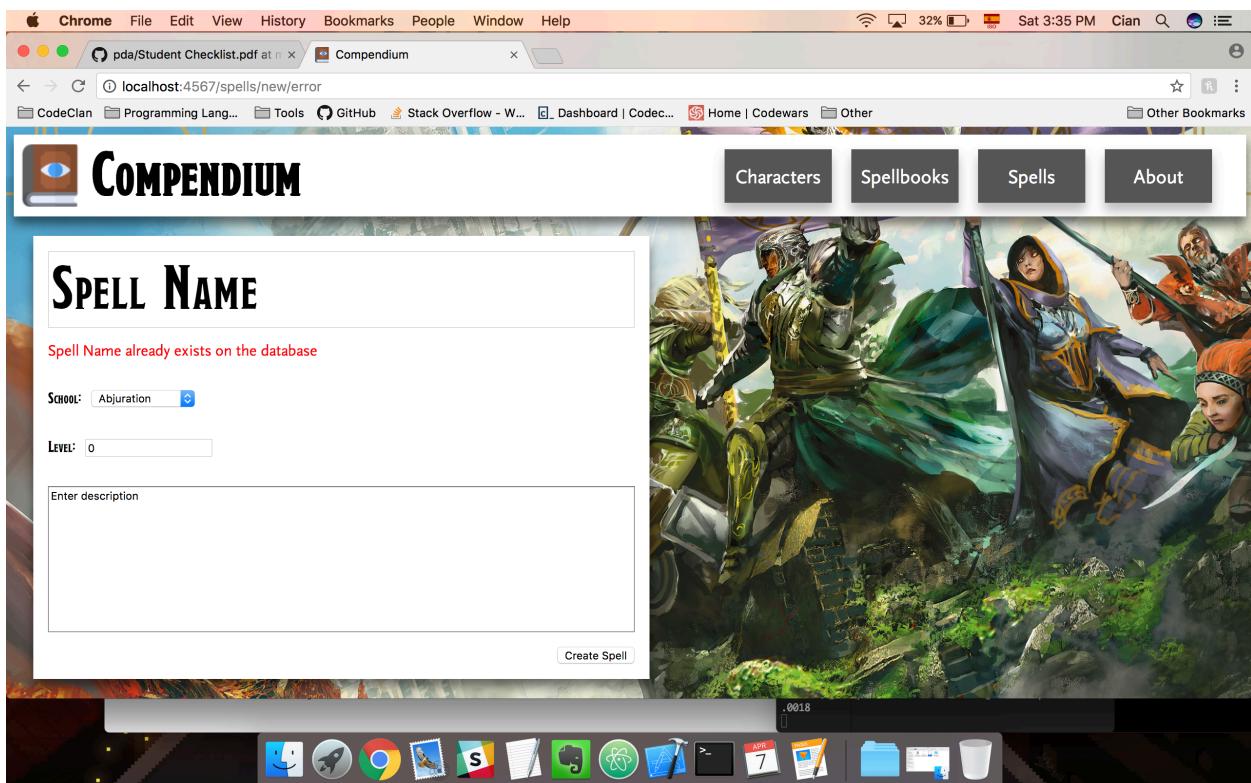




## P. 13 User input



User input is used, but returned an error message (Spell name already exists on the database)



## P. 14 Interaction with data persistence

App shows list of Characters. User click on Characters/New to add a new character.

The screenshot shows a web browser window titled "Compendium". The URL in the address bar is "localhost:4567/characters". The page features a header with the word "COMPENDIUM" next to a book icon. Below the header is a navigation bar with four buttons: "Characters", "Spellbooks", "Spells", and "About". The main content area displays a table with two rows of character data:

Class	Level	Name	Spellbooks	Description
Warlock	2	Gul'dan	3	Indebted to a demonic lord and pitted against the mentor he betrayed, Gul'dan is o...
Bard	1	Valtek	1	The bard uses music and magic to support and inspire the rest of the party. Bards c...

The background of the page is a large, vibrant illustration of a fantastical scene featuring a massive green dragon-like creature, a knight in armor, and other characters in a lush, rocky landscape.

The screenshot shows a web browser window titled "Compendium" with the URL "localhost:4567/characters/new". The page has a header with the word "COMPENDIUM" and a navigation bar with four buttons: "Characters", "Spellbooks", "Spells", and "About". The main content area contains a form for creating a new character:

**JAIME LOPEZ**

CHARACTER CLASS:

LEVEL:

Test of User Input being saved

The background of the page is a large, vibrant illustration of a fantastical scene featuring a massive green dragon-like creature, a knight in armor, and other characters in a lush, rocky landscape.

After clicking on create character, the list show the previous character but now new character has been added.

The screenshot shows a web browser window with the title bar "pda/Student Checklist.pdf at in" and the tab "Compendium". The address bar shows "localhost:4567/characters". The page content is titled "Compendium" with a logo of a book with an eye. It features a navigation bar with four buttons: "Characters", "Spellbooks", "Spells", and "About". Below the navigation bar is a table with columns: Class, Level, Name, Spellbooks, and Description. The table contains three rows:

Class	Level	Name	Spellbooks	Description
Warlock	2	Gul'dan	3	Indebted to a demonic lord and pitted against the mentor he betrayed, Gul'dan is o...
Bard	1	Jaime Lopez	0	Test of User Input being saved
Bard	1	Valtek	1	The bard uses music and magic to support and inspire the rest of the party. Bards c...

The background of the page features a large, detailed fantasy illustration of a city under attack, with a dragon breathing fire and various characters in the foreground.

## P. 15 User output result

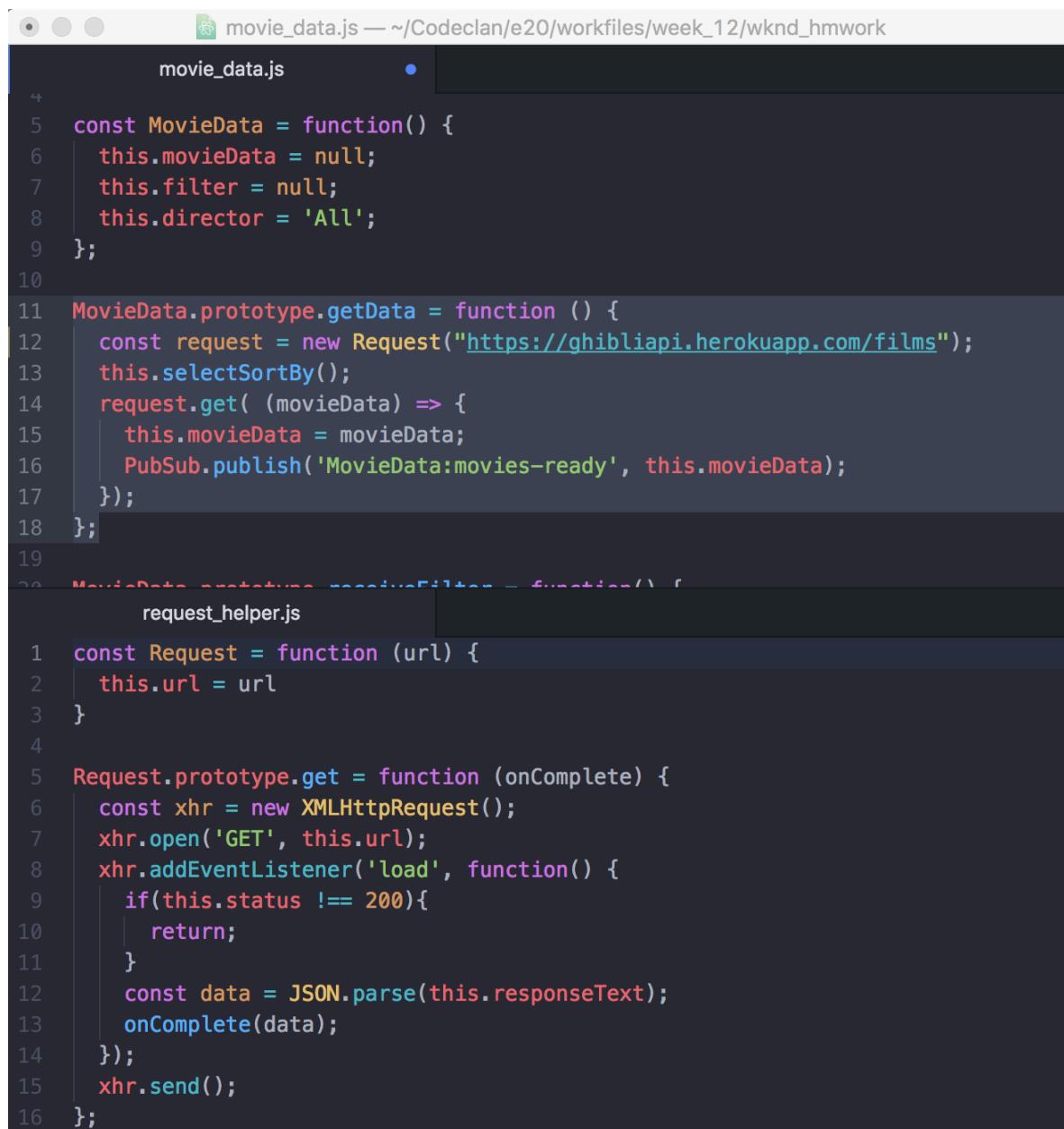
The screenshot shows a web application titled "Compendium". On the left, under "SPELLS", the "MAGIC STONE" spell is displayed. It's a Transmutation Level 1 spell. The description states: "You touch one to three pebbles and imbue them with magic. You or someone else can make a ranged spell attack with one of the pebbles by throwing it or hurling it with a sling. If thrown, it has a range of 60 feet. If someone else attacks with the pebble, that attacker adds your spellcasting ability modifier, not the attacker's, to the attack roll. On a hit, the target takes bludgeoning damage equal to 1d6 + your spellcasting ability modifier. Hit or miss, the spell then ends on the stone. If you cast this spell again, the spell ends early on any pebbles still affected by it." Below the spell description are buttons for "Go Back", "Edit Spell", and "Delete Spell". A "LEARN" section allows adding the spell to a student's spellbook. On the right, under "STUDENTS", a table lists "Gul'dan" with "Evocation spells" in their spellbook. There is a "Delete" button next to Gul'dan's entry. The browser's toolbar at the top includes icons for file operations, history, and search.

User click on delete to delete and entry on the “Students” table, and website refreshes showing that it was deleted. P. 16 Show an API being used within your program.

This screenshot shows the same web application after the "Delete" operation for Gul'dan has been performed. The "STUDENTS" table now shows an empty row for Gul'dan, indicating that the entry has been removed. The rest of the interface remains the same, displaying the "MAGIC STONE" spell and its details. The browser's toolbar at the top is visible.

## 16. Show an API being used within your program.

The code that uses or implements the API:



```
movie_data.js — ~/Codeclan/e20/workfiles/week_12/wknd_hmwork
  4
  5 const MovieData = function() {
  6   this.movieData = null;
  7   this.filter = null;
  8   this.director = 'All';
  9 }
 10
 11 MovieData.prototype.getData = function () {
 12   const request = new Request("https://ghibliapi.herokuapp.com/films");
 13   this.selectSortBy();
 14   request.get( (movieData) => {
 15     this.movieData = movieData;
 16     PubSub.publish('MovieData:movies-ready', this.movieData);
 17   });
 18 }
 19
 20 MovieData.prototype.selectSortBy = function() {
 21   request_helper.js
 22
 23   request_helper.js
 24
 25   const Request = function (url) {
 26     this.url = url
 27   }
 28
 29   Request.prototype.get = function (onComplete) {
 30     const xhr = new XMLHttpRequest();
 31     xhr.open('GET', this.url);
 32     xhr.addEventListener('load', function() {
 33       if(this.status !== 200){
 34         return;
 35       }
 36       const data = JSON.parse(this.responseText);
 37       onComplete(data);
 38     });
 39     xhr.send();
 40   };
 41 }
```

## The API being used by the program whilst running:

Filter by Director  Sort by

 Castle in the Sky 1986 The orphan Sheeta inherited a mysterious crystal	 Grave of the Fireflies 1988 In the latter part of World War II, a boy and his	 My Neighbor Totoro 1988 Two sisters move to the country with their father
--	--	--

**P. 17 Bug tracking report showing the errors diagnosed and corrected.**

Bug tracking report			
Player number must be a number between 1 and 4	Pass	Questions number must be a number between 5 and 50	Pass
When timer stops, player turn must be skipped and question should be marked as incorrect	Pass	When player answers correctly, question is added to both array of correct answers and array of all questions answered	Pass
When player answers, timer must stop	Pass	Player must not be able to click on any answer when correct answer is shown and next question is being prepared	Fail
Only questions from selected categories and difficulty are loaded from the API	Pass	Timer must only be created when new round or turn is initiated	Fail

## P.18 Testing your program.

```
card.rb
1
2 class Card
3   attr_reader :suit, :value
4
5   def initialize(suit, value)
6     @suit = suit
7     @value = value;
8   end
9
10  def checkforAce(card)
11    if card.value == 1
12      return true
13    else
14      return false
15    end
16  end
17
18  def self.highest_card(card1, card2)
19    if card1.value > card2.value
20      return card1.name
21    else
22      card2
23    end
24  end
25
26  def self.cards_total(cards)
27    total =
28    for card in cards
29      total += card.value
30    end
31    return "You have a total of" + total
32  end
33
34 end
```

```
card_spec.rb
1 require('minitest/autorun')
2 require('minitest/rng')
3 require_relative('../card')
4
5 class TestCard < MiniTest::Test
6   def setup
7     @card01 = Card.new("Hearts", 4)
8     @card02 = Card.new("Spades", 5)
9     @card03 = Card.new("Clubs", 8)
10    @card04 = Card.new("Diamonds", 1)
11    @cards = [@card01, @card02, @card03, @card04]
12  end
13
14  def test_checkforAce_returns_true()
15    result = Card.checkforAce(@card04)
16    assert_equal(true, result)
17  end
18
19  def test_checkforAce_returns_false()
20    result = Card.checkforAce(@card01)
21    assert_equal(false, result)
22  end
23
24  def test_highest_card()
25    result = Card.highest_card(@card01, @card02)
26    assert_equal("Spades", result.suit)
27  end
28
29  def test_cards_total()
30    result = Card.cards_total(@cards)
31    assert_equal(19, result)
32  end
33
```

```
specs ruby card_spec.rb
Run options: --seed 31039

# Running:

E

Error:
TestCard#test_cards_total:
NameError: undefined local variable or method `total' for
Card:Class
  /Users/user/Codeclan/e20/workfiles/PDA files/Dynamic T
esting/Testing_task_2/card.rb:27:in `cards_total'
    card_spec.rb:30:in `test_cards_total'

bin/rails test card_spec.rb:29

E

Error:
TestCard#test_checkforAce_returns_false:
NoMethodError: undefined method `checkforAce' for Card:Cl
ass
  card_spec.rb:20:in `test_checkforAce_returns_false'

bin/rails test card_spec.rb:19

E

Error:
TestCard#test_checkforAce_returns_true:
NoMethodError: undefined method `checkforAce' for Card:Cl
ass
  card_spec.rb:15:in `test_checkforAce_returns_true'

bin/rails test card_spec.rb:14

.
.

Finished in 0.008625s, 463.7681 runs/s, 115.9420 assertion
s/s.
4 runs, 1 assertions, 0 failures, 3 errors, 0 skips
→ specs
```

```
card.rb
1
2 def initialize(suit, value)
3   @suit = suit
4   @value = value;
5 end
6
7 def self.checkforAce(card)
8   if card.value == 1
9     return true
10  else
11    return false
12  end
13
14  def self.highest_card(card1, card2)
15    if card1.value > card2.value
16      return card1
17    elsif card2.value > card1.value
18      return card2
19    else
20      return "Same Value"
21    end
22  end
23
24  def self.cards_total(cards)
25    total = 0
26    for card in cards
27      total += card.value
28    end
29    return total
30  end
31
32  end
33
34 end
```

```
card_spec.rb
1 require('minitest/autorun')
2 require('minitest/rng')
3 require_relative('../card')
4
5 class TestCard < MiniTest::Test
6   def setup
7     @card01 = Card.new("Hearts", 4)
8     @card02 = Card.new("Spades", 5)
9     @card03 = Card.new("Clubs", 8)
10    @card04 = Card.new("Diamonds", 1)
11    @cards = [@card01, @card02, @card03, @card04]
12  end
13
14  def test_checkforAce_returns_true()
15    result = Card.checkforAce(@card04)
16    assert_equal(true, result)
17  end
18
19  def test_checkforAce_returns_false()
20    result = Card.checkforAce(@card01)
21    assert_equal(false, result)
22  end
23
24  def test_highest_card()
25    result = Card.highest_card(@card01, @card02)
26    assert_equal("Spades", result.suit)
27  end
28
29  def test_cards_total()
30    result = Card.cards_total(@cards)
31    assert_equal(18, result)
32  end
33
```

```
specs ruby card_spec.rb
Run options: --seed 13120

# Running:
.

.

Finished in 0.001504s, 2659.5745 runs/s, 2659.5745 asserti
ons/s.
4 runs, 4 assertions, 0 failures, 0 errors, 0 skips
→ specs
```