

Evidence for Project Unit

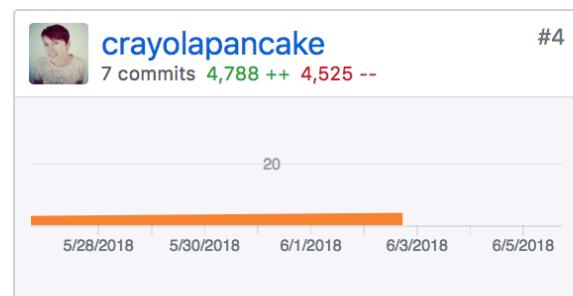
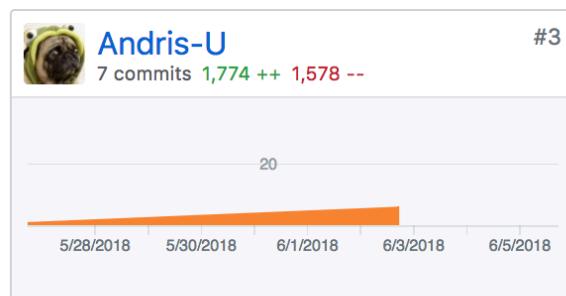
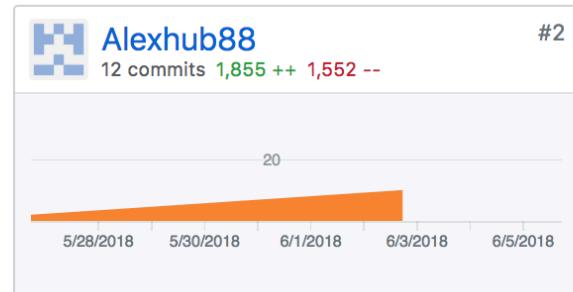
Jaime Lopez
Cohort E20

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

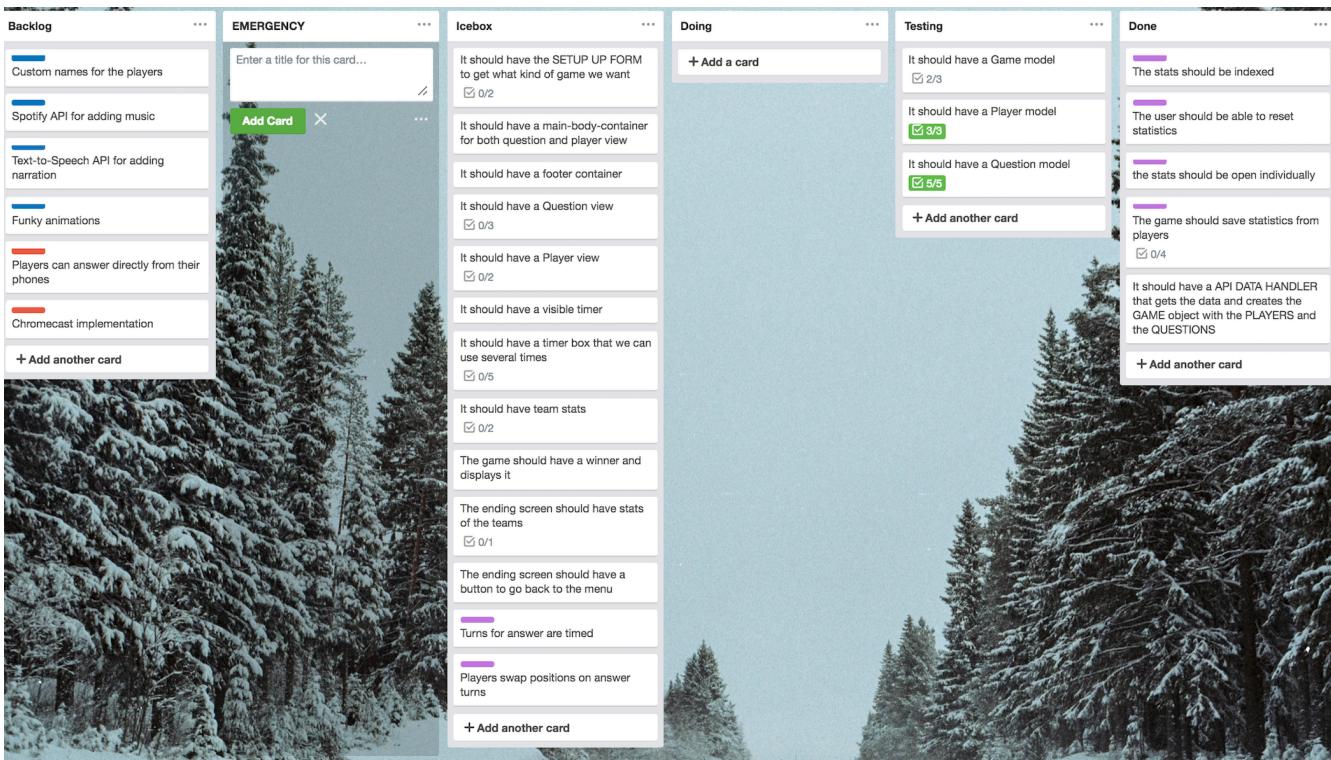
Extensions

- 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

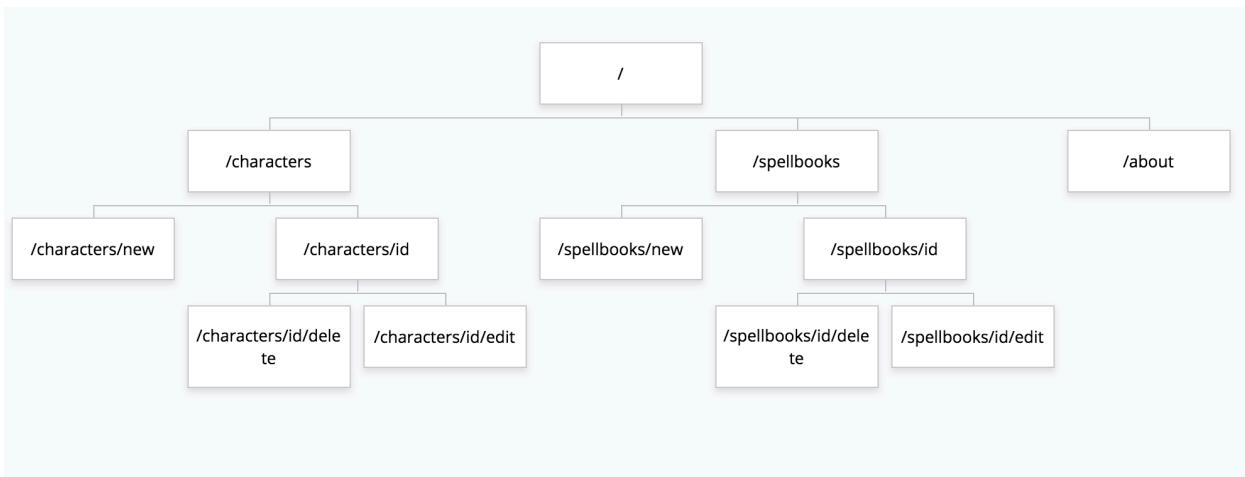
- <https://www.highcharts.com/> HighCharts is an open-source library for rendering responsive charts.
- <https://opentdb.com/> OpenTriviaDB is an open-source library of trivia questions.

P. 3 Use of Trello



P. 4 Acceptance Criteria

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



P. 7 System interactions diagrams

P. 8 Two Object Diagrams

Evidence for unit

Evidence for unit

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

On this example please take a screenshot and write what it is doing and why you decided to use it.

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

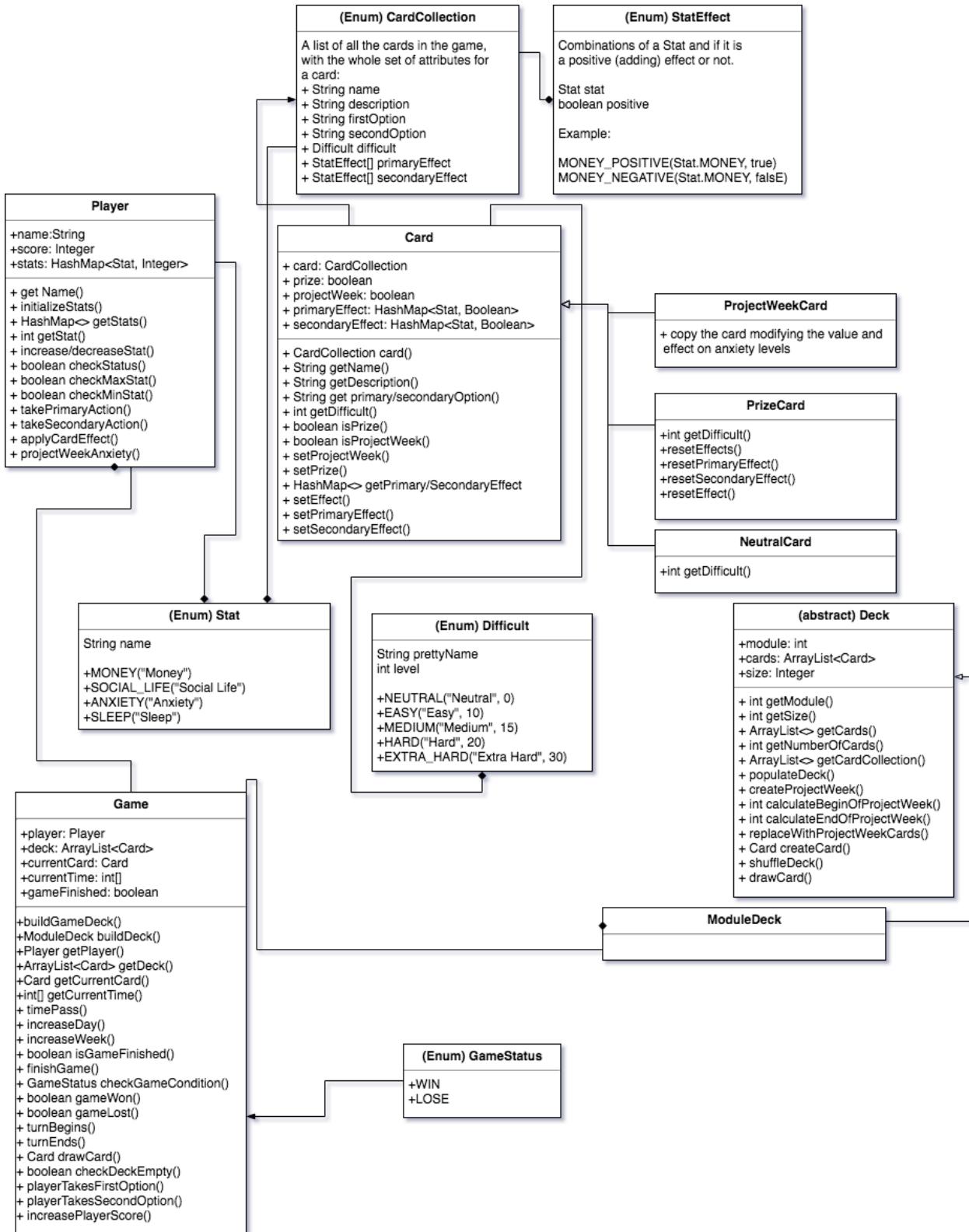
Add topics

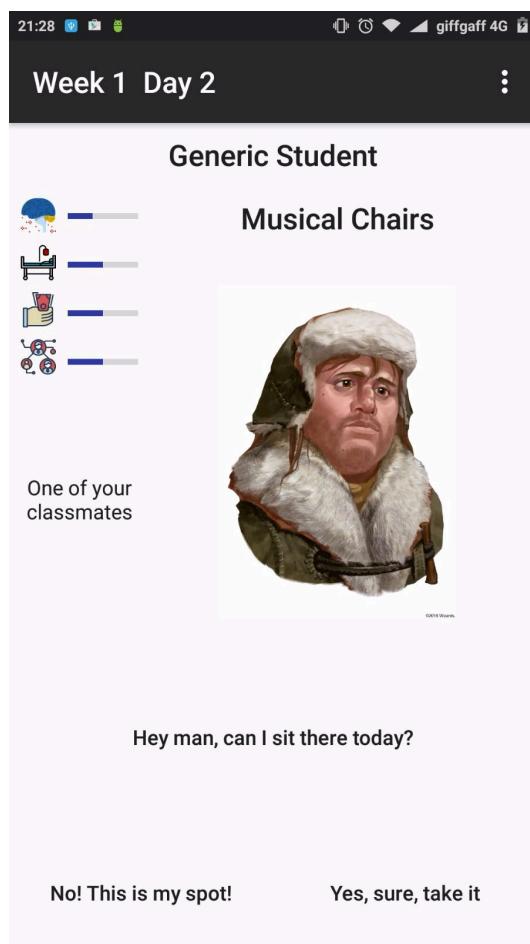
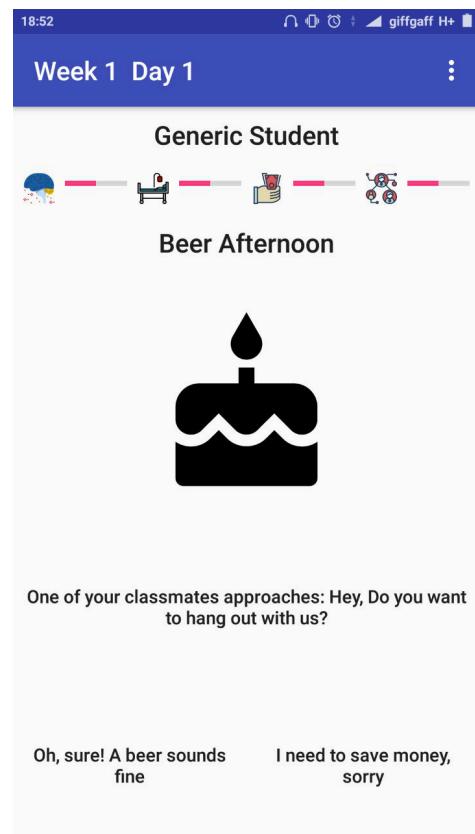
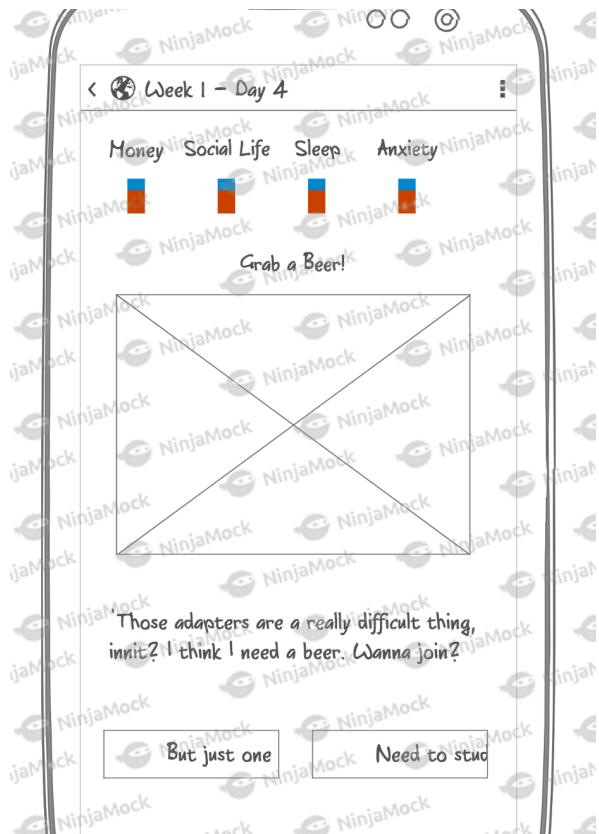
103 commits | 1 branch | 0 releases | 1 contributor

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

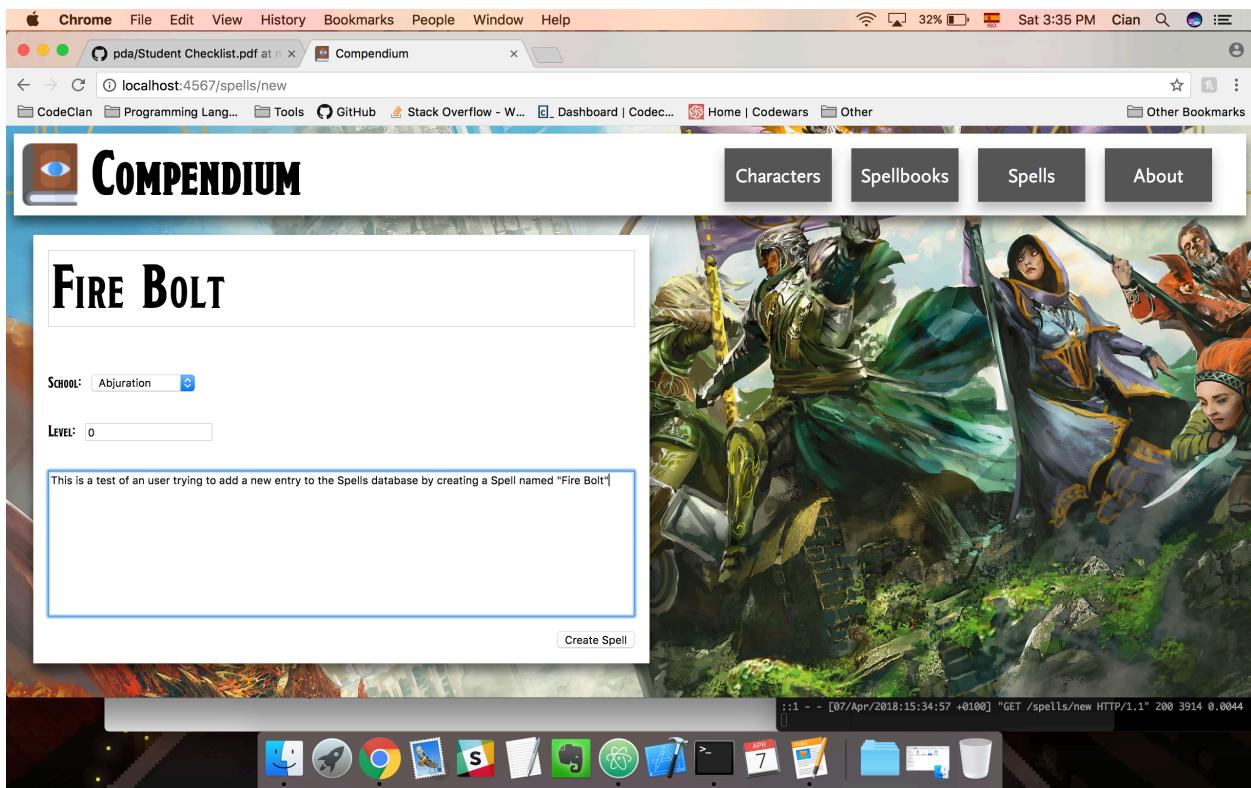
File	Description	Time Ago
DetectiveAzul Showing breaks on spell show.erb	Latest commit 7fb27d7 14 days ago	
controllers	Added 404 website	22 days ago
db	Fixed sql	14 days ago
models	fixed error on method to cheack if spell is repeated on a book	23 days ago
public	Showing breaks on spell show.erb	14 days 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.

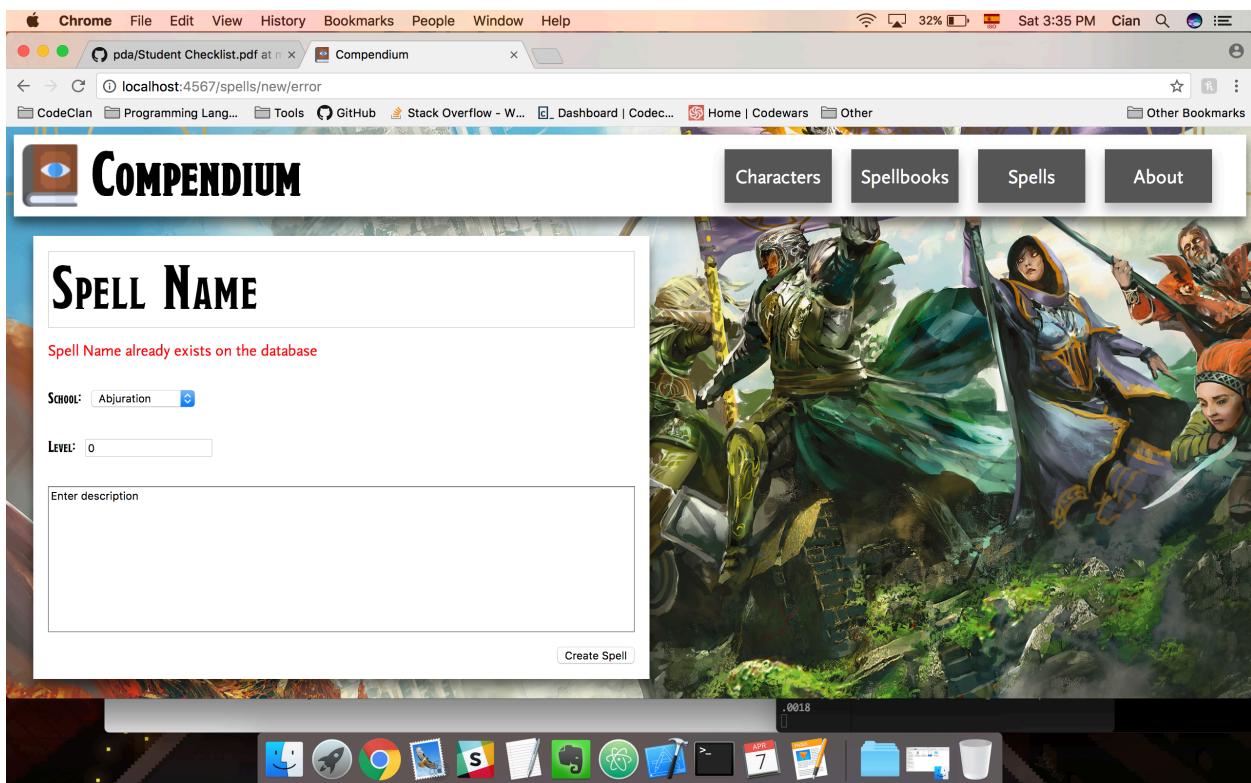




P. 13 User input



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



P. 14 Interaction with data persistence

The screenshot shows a web browser window titled "Compendium" at the URL 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 columns: Class, Level, Name, Spellbooks, and Description. Two character entries are visible:

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 dragon, a knight in armor, and other characters.

The screenshot shows a web browser window titled "Compendium" at the URL localhost:4567/characters/new. The page has a similar layout to the first one, with the "COMPENDIUM" header and navigation bar. 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 the same vibrant, fantastical illustration as the first screenshot.

localhost:4567/characters

CodeClan Programming Lang... Tools GitHub Stack Overflow - W... Dashboard | Codec... Home | Codewars Other Other Bookmarks

Compendium

Characters Spellbooks Spells About

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...

localhost:4567/characters/12

P. 15 User output result

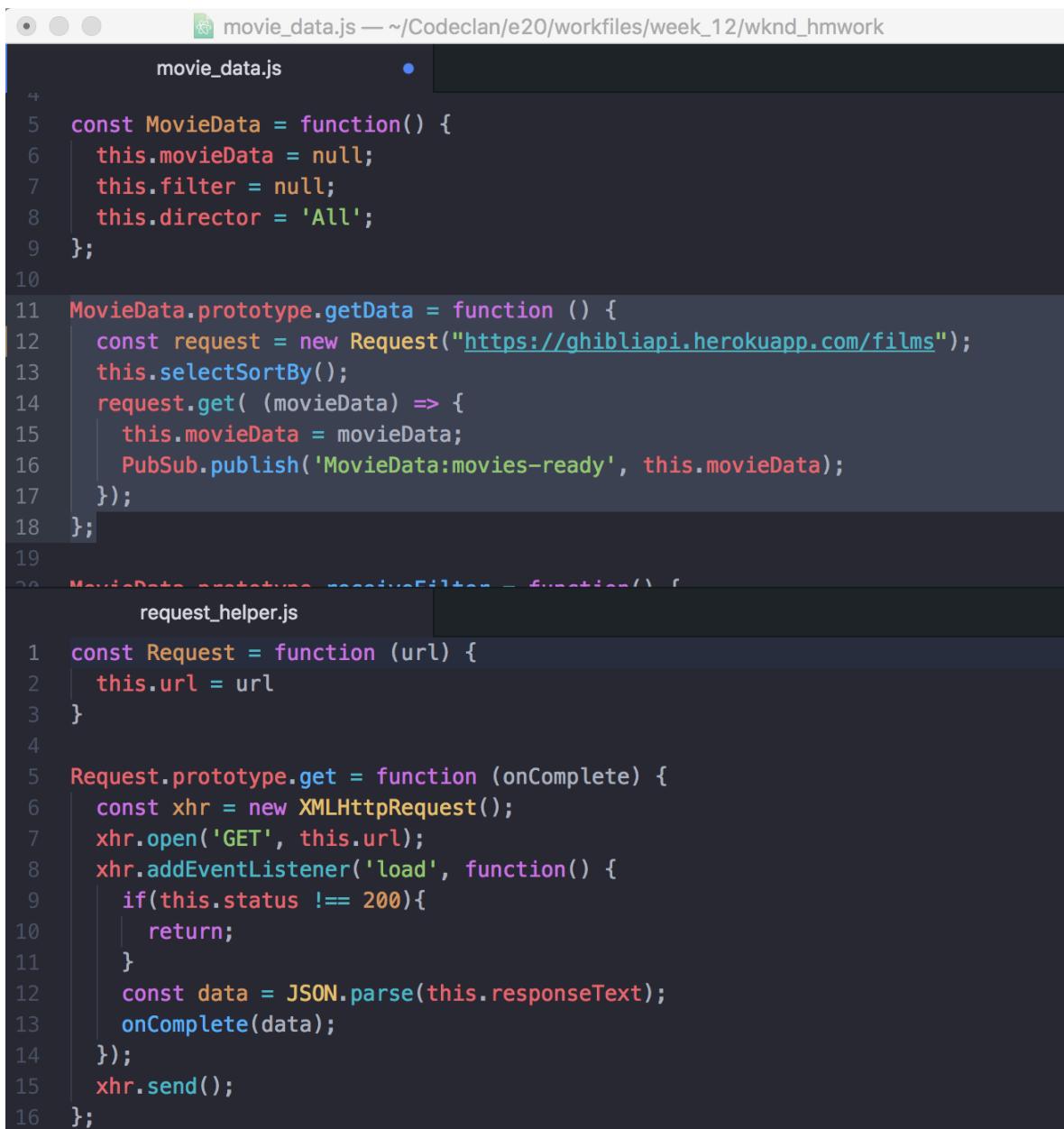
User click on delete to delete and entry on the “Students” table, and website refreshes showing that it was deleted.

The screenshot shows a web browser window for the 'Compendium' application. The main content area displays a spell card for 'MAGIC STONE' at 'TRANSMUTATION LEVEL 1'. Below the spell card is a 'LEARN' section. To the right, a 'STUDENTS' table lists a single entry: 'Gul'dan' under 'Student' and 'Evocation spells' under 'Spellbook'. A 'Delete' button is visible next to the entry. The browser's address bar shows 'localhost:4567/spells/4'. The top navigation bar includes links for Characters, Spellbooks, Spells, and About. The bottom of the screen shows a Mac OS X desktop interface with various application icons in the Dock.

This screenshot is identical to the one above, showing the 'Compendium' application. The 'STUDENTS' table now shows no entries, indicating that the 'Delete' operation has been completed. The rest of the interface, including the spell card, learn section, and browser status, remains the same.

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

The code that uses or implements the API:



The screenshot shows a terminal window with two tabs open. The active tab is 'movie_data.js' which contains the following code:

```
movie_data.js — ~/Codeclan/e20/workfiles/week_12/wknd_hmwork
movie_data.js      •

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.getFiltered = function() {
request_helper.js
```

The second tab is 'request_helper.js' which contains the following code:

```
const Request = function (url) {
  this.url = url
}

Request.prototype.get = function (onComplete) {
  const xhr = new XMLHttpRequest();
  xhr.open('GET', this.url);
  xhr.addEventListener('load', function() {
    if(this.status !== 200){
      return;
    }
    const data = JSON.parse(this.responseText);
    onComplete(data);
  });
  xhr.send();
};
```

The API being used by the program whilst running:

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

P. 18 Testing your program

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

Testing_task_2/card.rb 10:7

bin/rails test card_spec.rb:29

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:19

E

Error:

TestCard#test_checkforAce__returns_false:
NoMethodError: undefined method `checkForAce' for Card:Cla
ss
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:Cla
ss
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.
4 runs, 1 assertions, 0 failures, 3 errors, 0 skips

→ specs

```
card.rb
1 class Card
2   attr_reader :suit
3   @value = value;
4
5   def initialize(suit, value)
6     @suit = suit
7     @value = value;
8   end
9
10  def self.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
21    elsif card2.value > card1.value
22      return card2
23    else
24      return "Same Value"
25    end
26  end
27
28  def self.cards_total(cards)
29    total = 0
30    for card in cards
31      total += card.value
32    end
33    return total
34  end
35
36 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
```

Testing_task_2/specs/card_spec.rb 31:20

bin/rails test card_spec.rb:29

E

...

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

→ specs