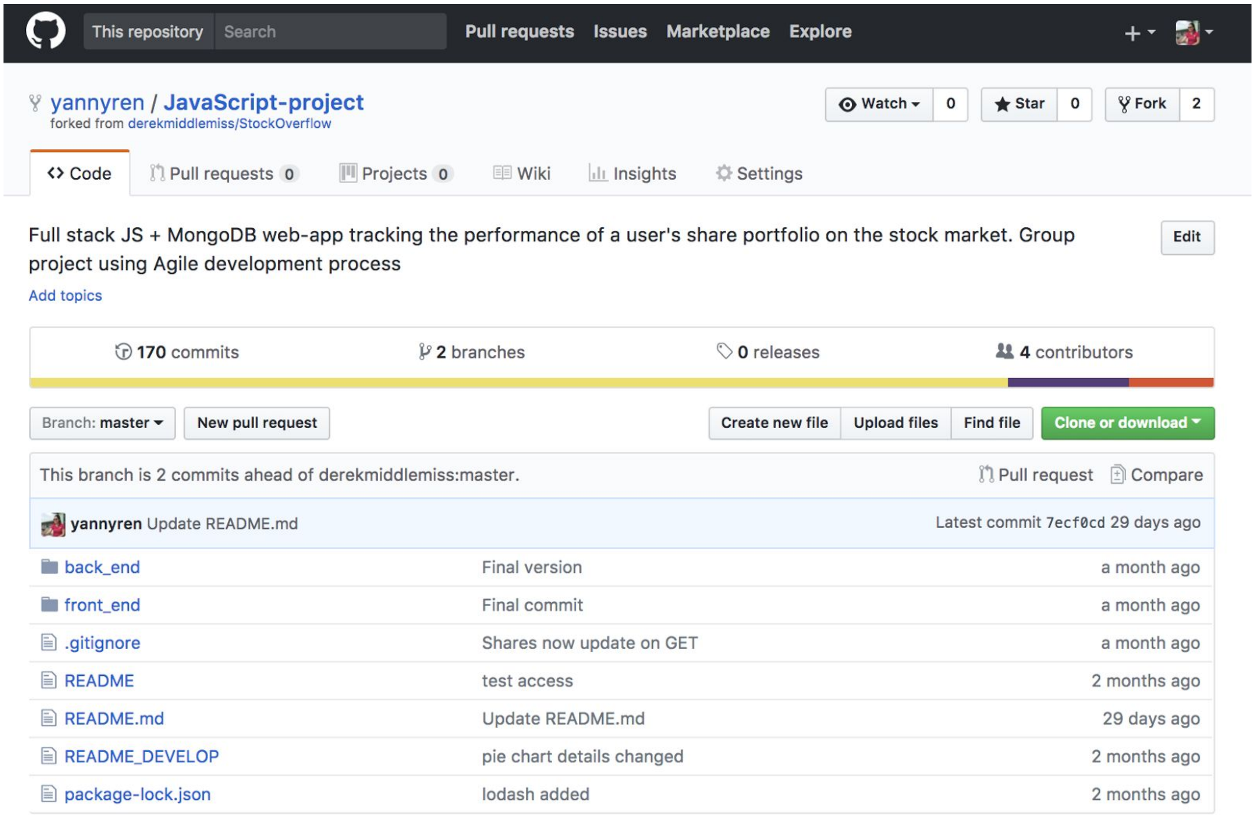
Project Unit
SQA PDA: Software Development
Yan Ren

P1 - Group Project Github



yannyren / JavaScript-project
forked from derekmiddlemiss/StockOverflow

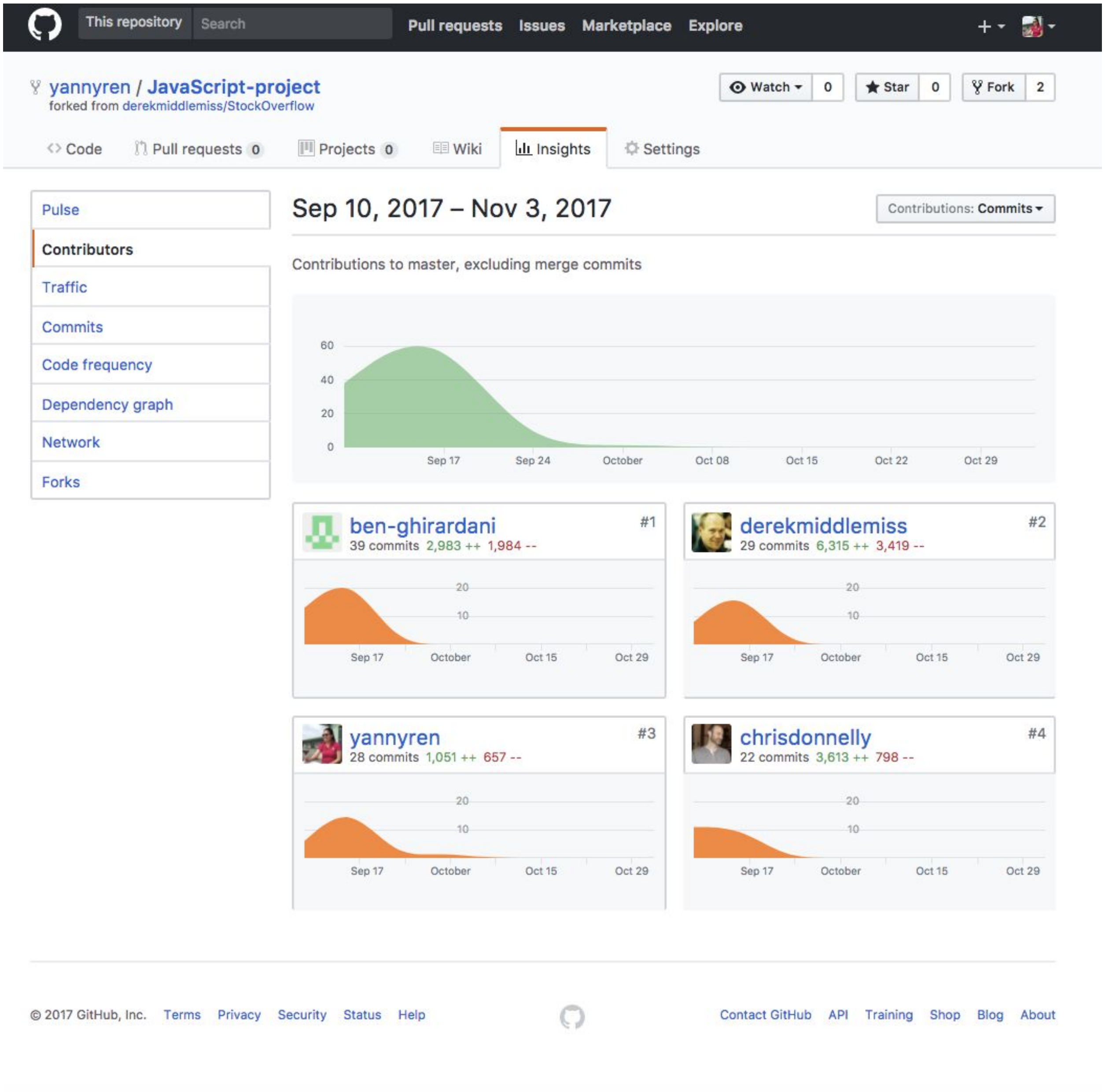
Full stack JS + MongoDB web-app tracking the performance of a user's share portfolio on the stock market. Group project using Agile development process

170 commits 2 branches 0 releases 4 contributors

This branch is 2 commits ahead of derekmiddlemiss:master.

yannyren Update README.md Latest commit 7ecf0cd 29 days ago

back_end	Final version	a month ago
front_end	Final commit	a month ago
.gitignore	Shares now update on GET	a month ago
README	test access	2 months ago
README.md	Update README.md	29 days ago
README_DEVELOP	pie chart details changed	2 months ago
package-lock.json	lodash added	2 months ago



yannyren / JavaScript-project
forked from derekmiddlemiss/StockOverflow

- Pulse
- Contributors
- Traffic
- Commits
- Code frequency
- Dependency graph
- Network
- Forks

Sep 10, 2017 – Nov 3, 2017

Contributions to master, excluding merge commits

ben-ghirardani #1
39 commits 2,983 ++ 1,984 --

derekmiddlemiss #2
29 commits 6,315 ++ 3,419 --

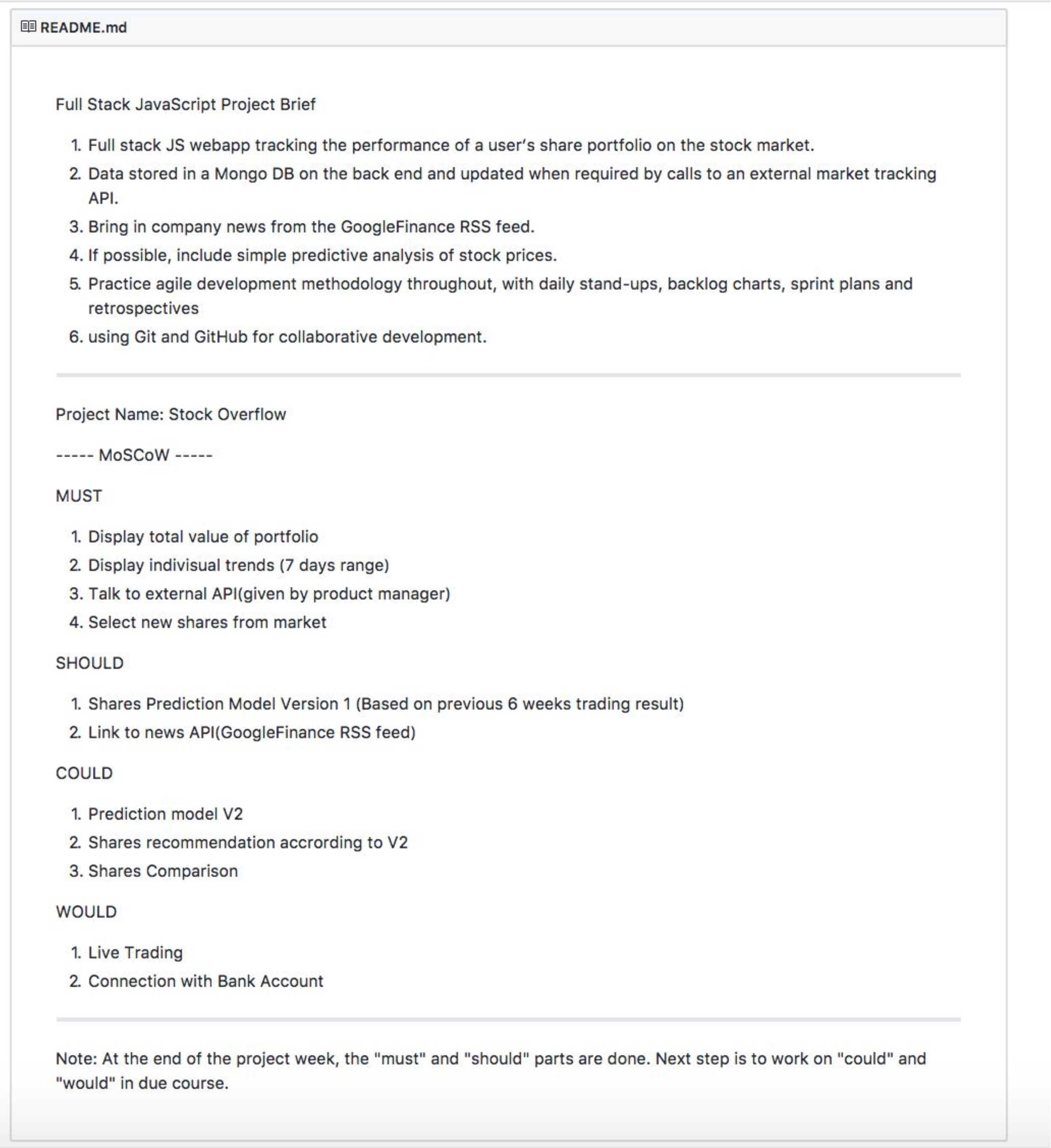
yannyren #3
28 commits 1,051 ++ 657 --

chrisdonnelly #4
22 commits 3,613 ++ 798 --

© 2017 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About

P2 - Group Project Brief



Full Stack JavaScript Project Brief

1. Full stack JS webapp tracking the performance of a user’s share portfolio on the stock market.
2. Data stored in a Mongo DB on the back end and updated when required by calls to an external market tracking API.
3. Bring in company news from the GoogleFinance RSS feed.
4. If possible, include simple predictive analysis of stock prices.
5. Practice agile development methodology throughout, with daily stand-ups, backlog charts, sprint plans and retrospectives
6. using Git and GitHub for collaborative development.

Project Name: Stock Overflow

----- MoSCoW -----

MUST

1. Display total value of portfolio
2. Display indivisual trends (7 days range)
3. Talk to external API(given by product manager)
4. Select new shares from market

SHOULD

1. Shares Prediction Model Version 1 (Based on previous 6 weeks trading result)
2. Link to news API(GoogleFinance RSS feed)

COULD

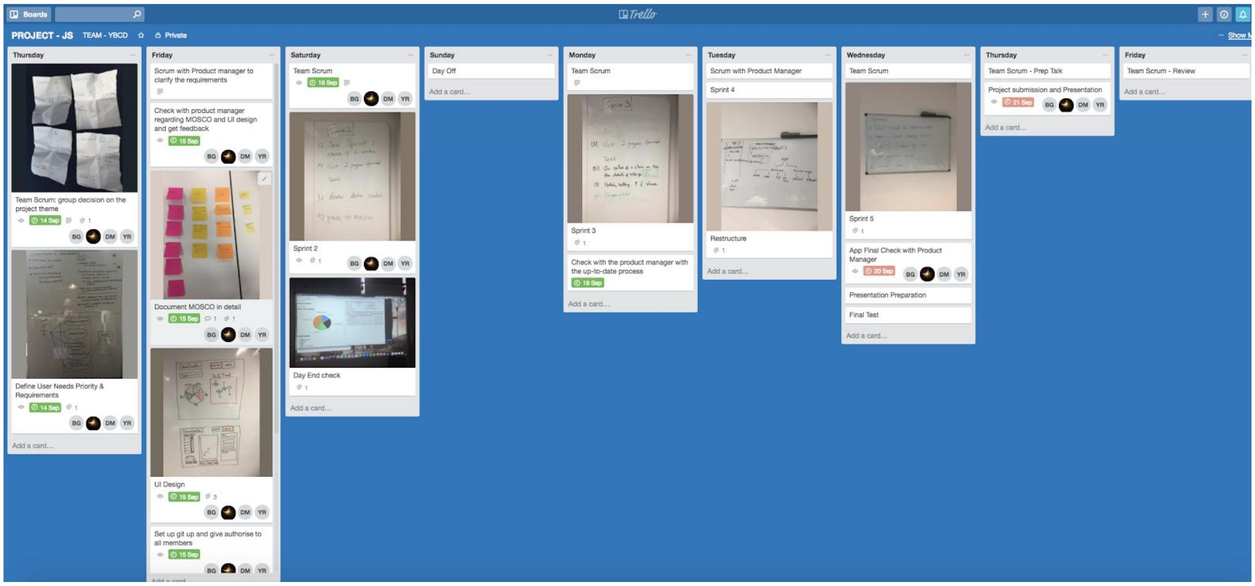
1. Prediction model V2
2. Shares recommendation accrording to V2
3. Shares Comparison

WOULD

1. Live Trading
2. Connection with Bank Account

Note: At the end of the project week, the "must" and "should" parts are done. Next step is to work on "could" and "would" in due course.

P3 - Group project planning



Boards

Trello

PROJECT - JS TEAM - YBCD Private

Show M

Thursday

- Team Scrum: group decision on the project theme — 14 Sep — 1 — BG DM YR
- Define User Needs Priority & Requirements — 14 Sep — 1 — BG DM YR

Add a card...

Friday

- Scrum with Product manager to clarify the requirements
- Check with product manager regarding MOSCO and UI design and get feedback — 15 Sep — BG DM YR
- Document MOSCO in detail — 15 Sep — 1 — 1 — BG DM YR
- UI Design — 15 Sep — 3 — BG DM YR
- Set up git up and give authorise to all members — 15 Sep — BG DM YR

Add a card...

Saturday

- Team Scrum — 16 Sep — BG DM YR
- Sprint 2 — 1 — BG DM YR
- Day End check — 1

Add a card...

Sunday

- Day Off

Add a card...

Monday

- Team Scrum
- Sprint 3 — 1
- Check with the product manager with the up-to-date process — 18 Sep

Add a card...

Tuesday

- Scrum with Product Manager
- Sprint 4
- Restructure — 1

Add a card...

Wednesday

- Team Scrum
- Sprint 5 — 1
- App Final Check with Product Manager — 20 Sep — BG DM YR
- Presentation Preparation
- Final Test

Add a card...

Thursday

- Team Scrum - Prep Talk
- Project submission and Presentation — 21 Sep — BG DM YR

Add a card...

Friday

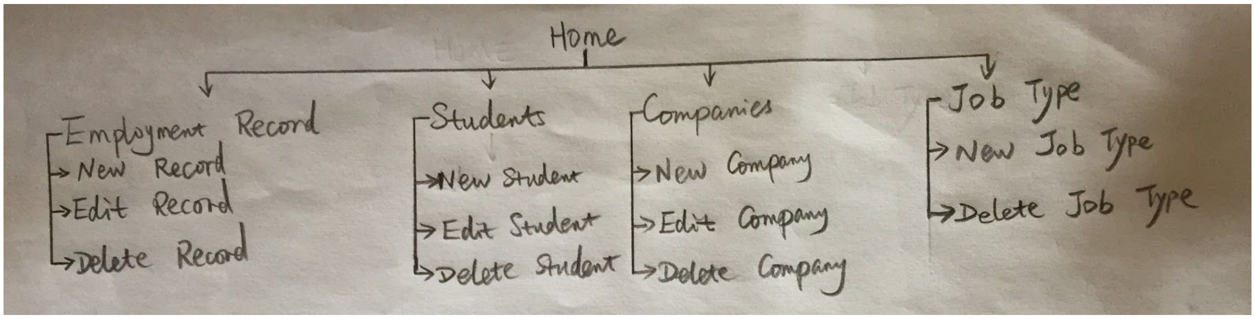
- Team Scrum - Review

Add a card...

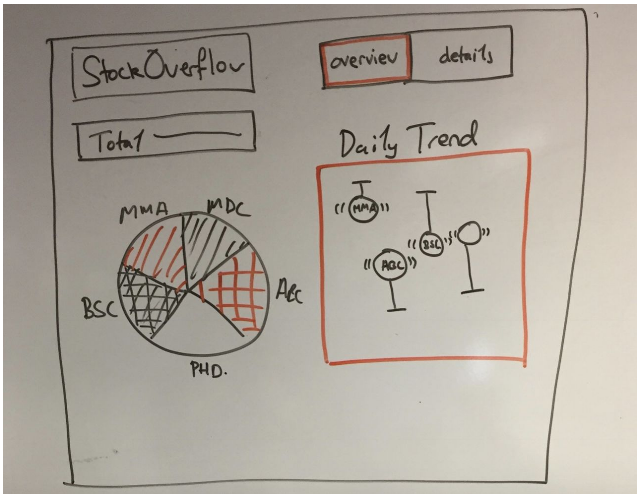
P4 - User acceptance and test plan (for final project)

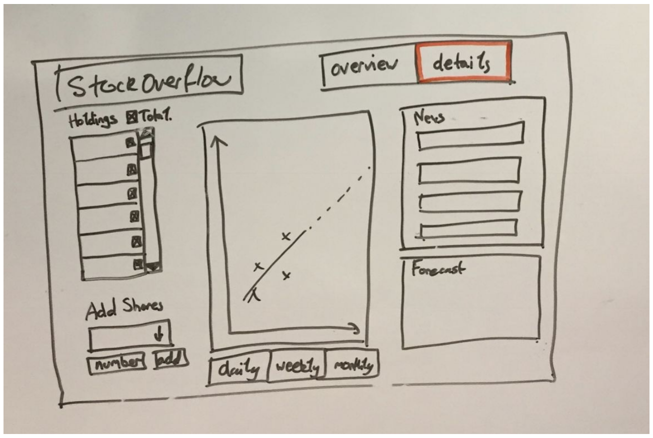
Acceptance Criteria	Expected Result/Output	Pass/Fail
A user can choose between the summary of the stock portfolio and the details page	User click on the switch icon on the righthand corner of the page	pass
A user can choose a stock	User input the stock number then search in the database. Once it is found, click on add button	pass
A user can see the related news of a chosen stock	User choose a stock on the leftside bar. The related recent news will be shown on the right handside column.	pass

P5 User Site Map

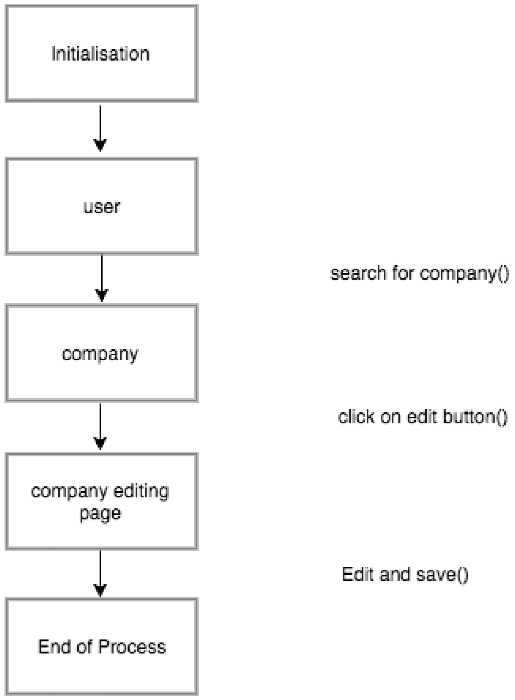


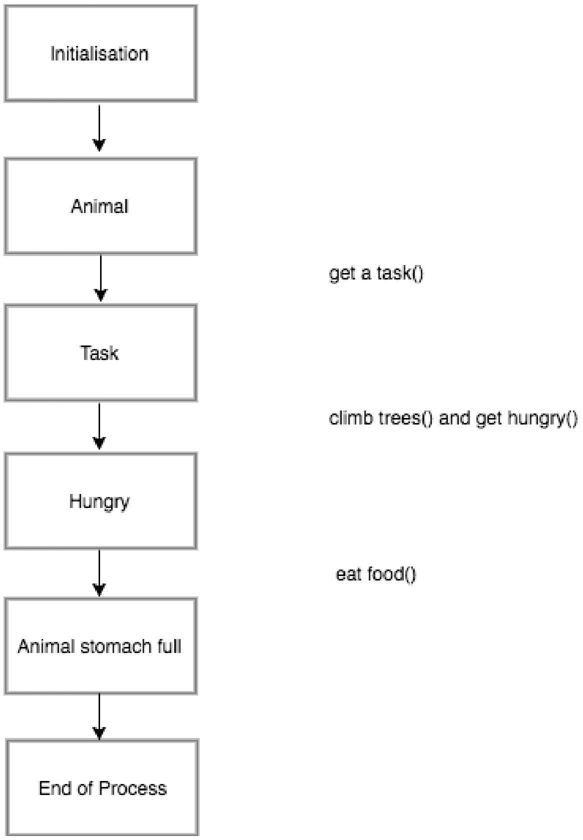
P6 - Wireframe Designs



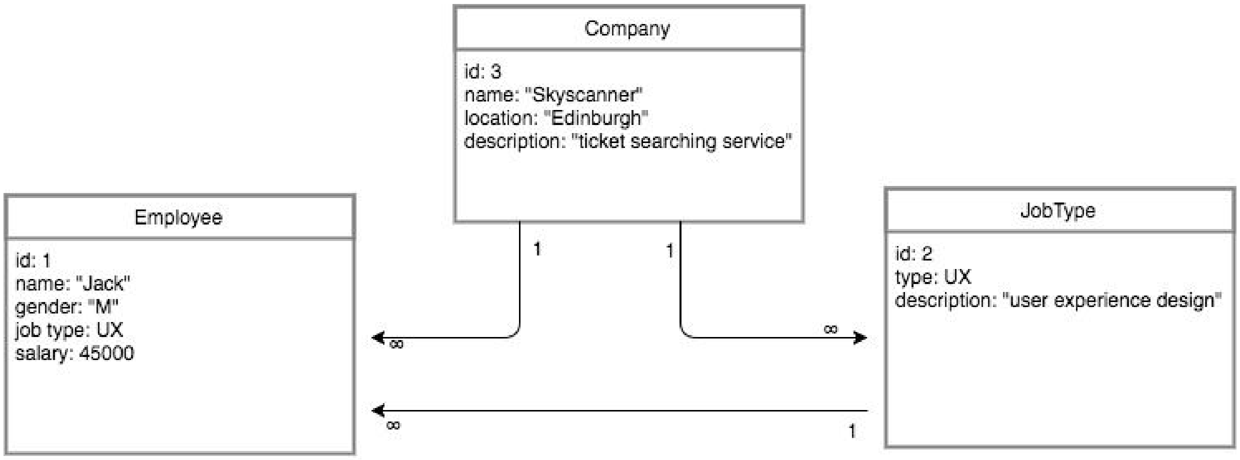


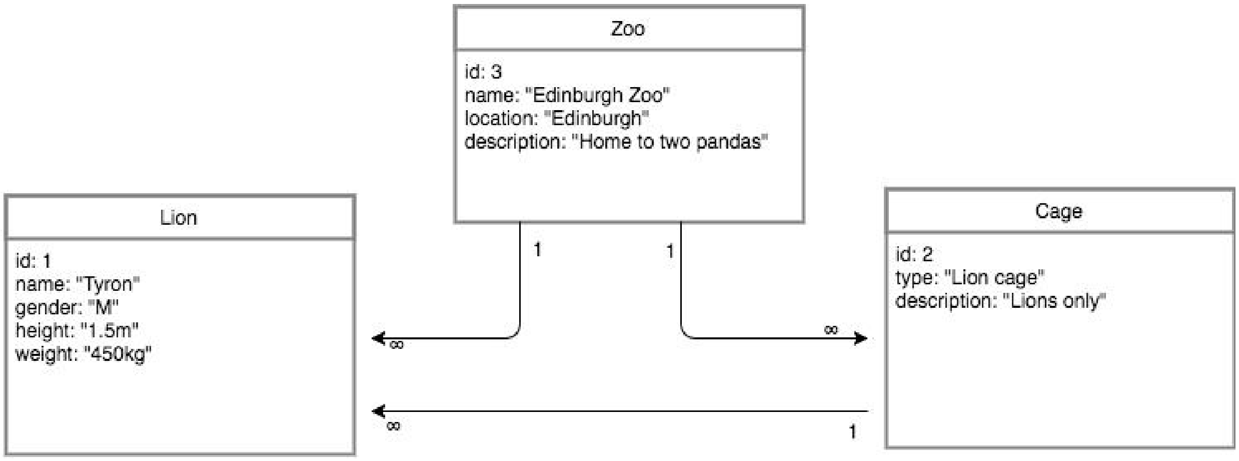
P7 - System Interaction Diagrams (Collaboration Diagrams)





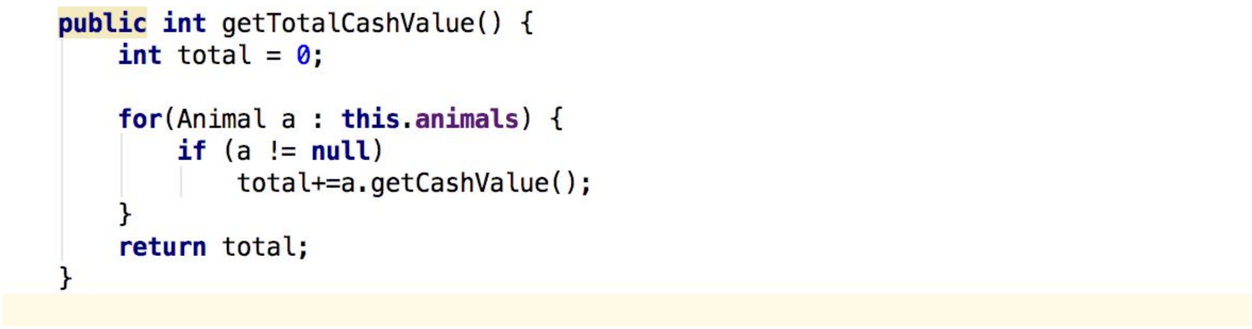
P8 Object Diagrams





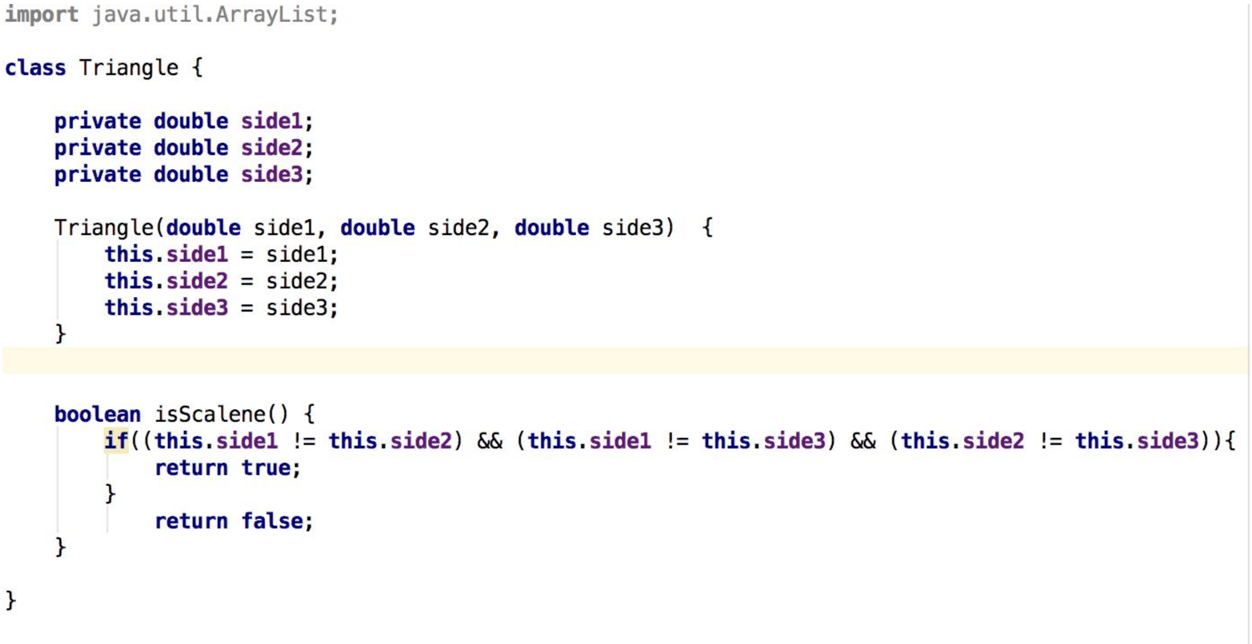
P9 Algorithms

In this algorithm, first we check if the animal object in the arraylist of animals is not null. If it is not null, we add the cash value of the animal to the total cash and then return the total.



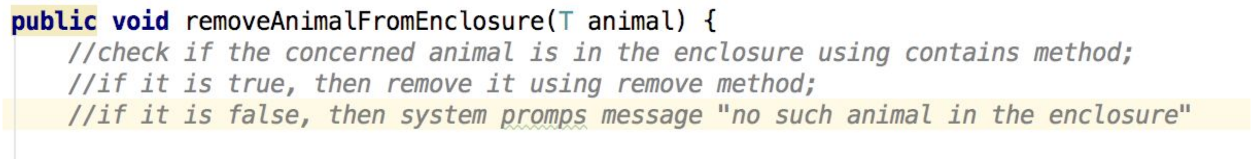
```
public int getTotalCashValue() {
    int total = 0;

    for(Animal a : this.animals) {
        if (a != null)
            total+=a.getCashValue();
    }
    return total;
}
```

In this algorithm, we want to check if all of the three sides of a triangle is not equal to each other.
If that is the case then the triangle in concern is a scalene. So we have to compare each side with the rest of the two and all results of the three comparisons will need to be false.



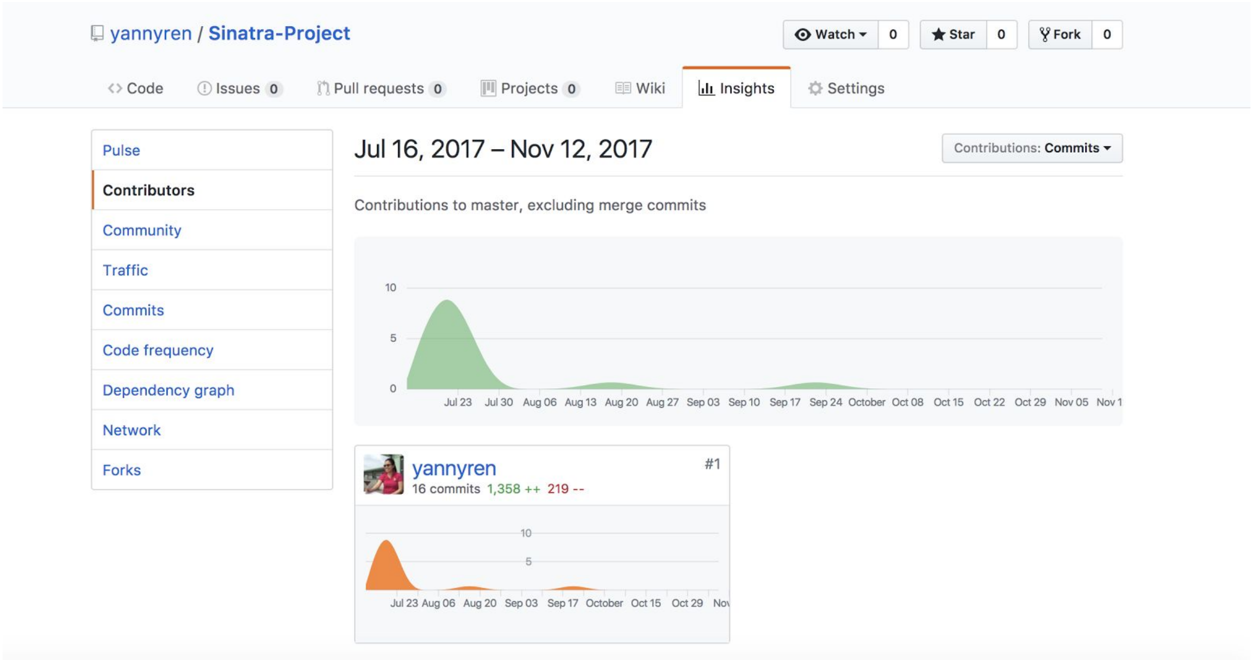
```
import java.util.ArrayList;

class Triangle {

    private double side1;
    private double side2;
    private double side3;

    Triangle(double side1, double side2, double side3)  {
        this.side1 = side1;
        this.side2 = side2;
        this.side3 = side3;
    }


    boolean isScalene() {
        if((this.side1 != this.side2) && (this.side1 != this.side3) && (this.side2 != this.side3)){
            return true;
        }
            return false;
    }

}
```

P10 Pseudocode



```
public void removeAnimalFromEnclosure(T animal) {
    //check if the concerned animal is in the enclosure using contains method;
    //if it is true, then remove it using remove method;
    //if it is false, then system promps message "no such animal in the enclosure"
```

P11 Project in Github



yannyren / Sinatra-Project

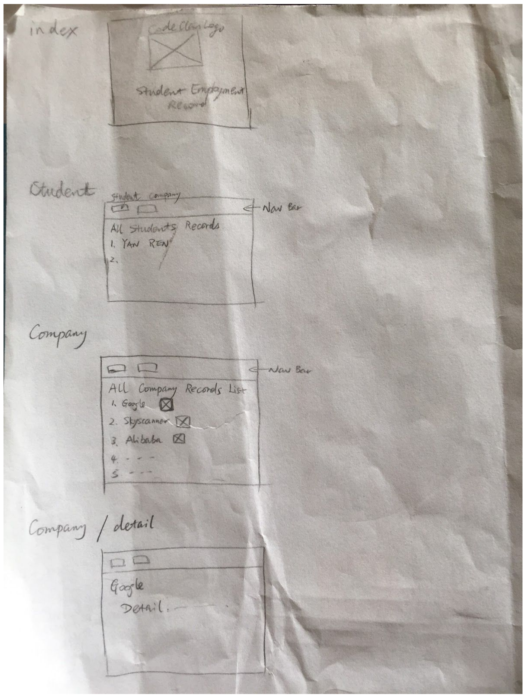
Jul 16, 2017 – Nov 12, 2017

Contributions to master, excluding merge commits

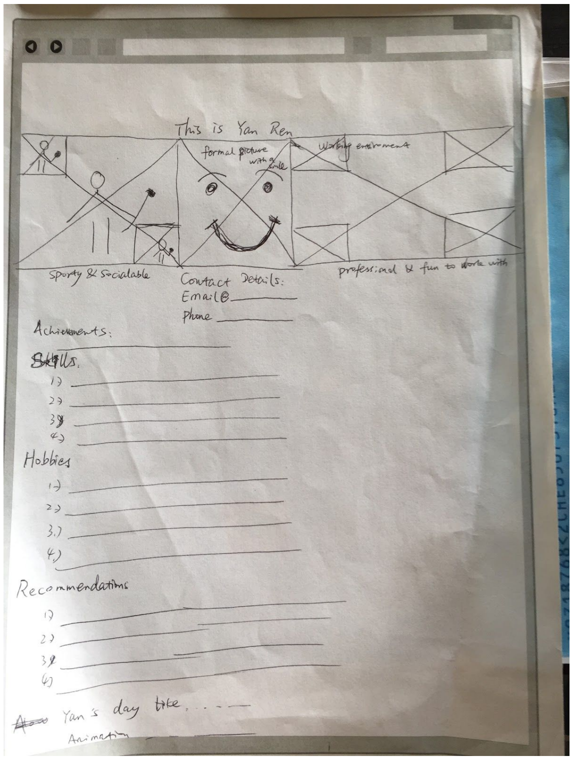
yannyren #1
16 commits 1,358 ++ 219 --

P12 Planning stage

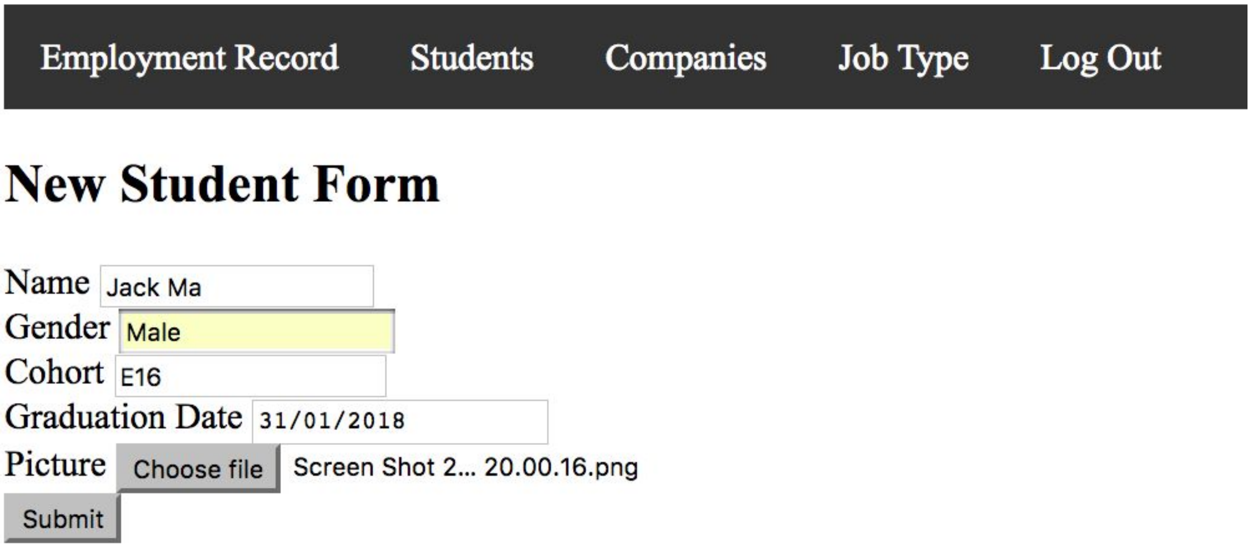
Stage 1



Stage 2



P13 Show user input being processed according to design requirements:



New Student Form

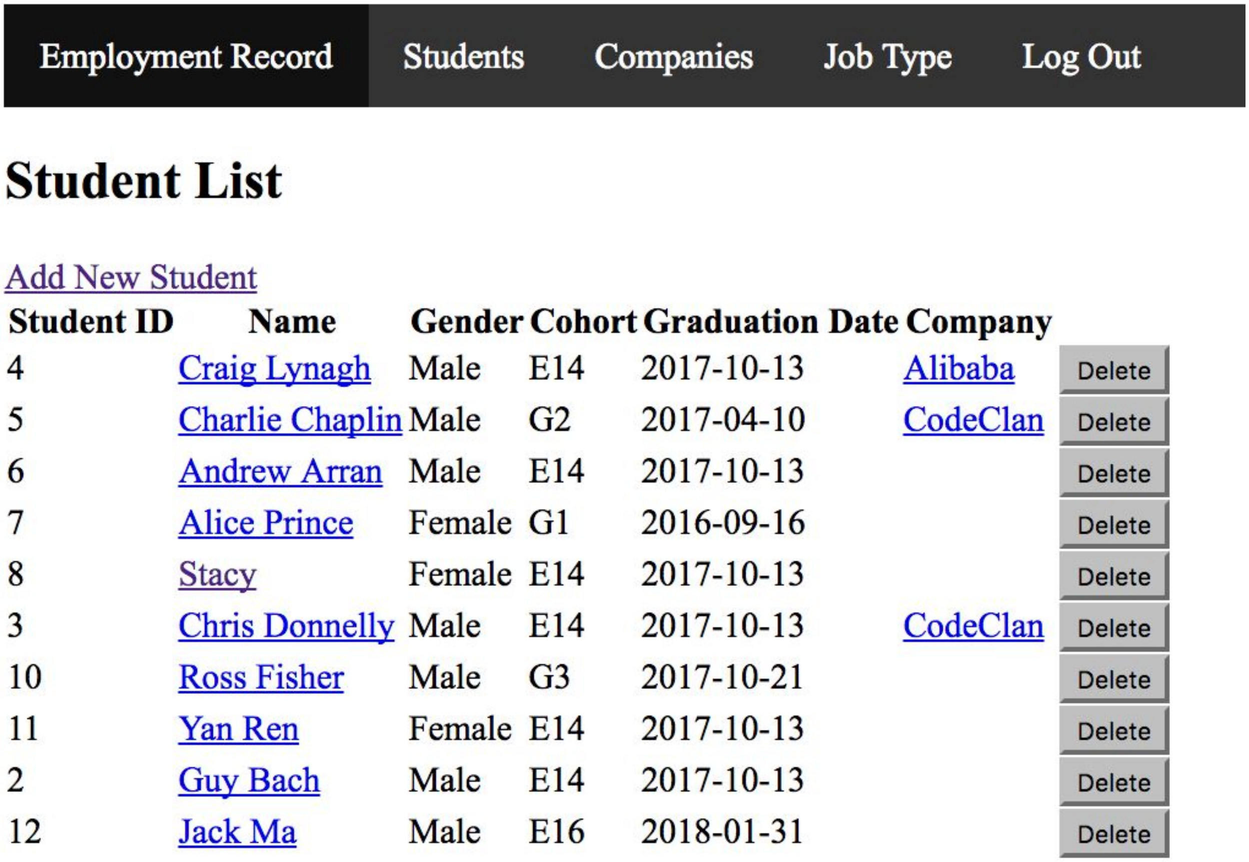
Name Jack Ma

Gender Male

Cohort E16

Graduation Date 31/01/2018

Picture Choose file Screen Shot 2... 20.00.16.png

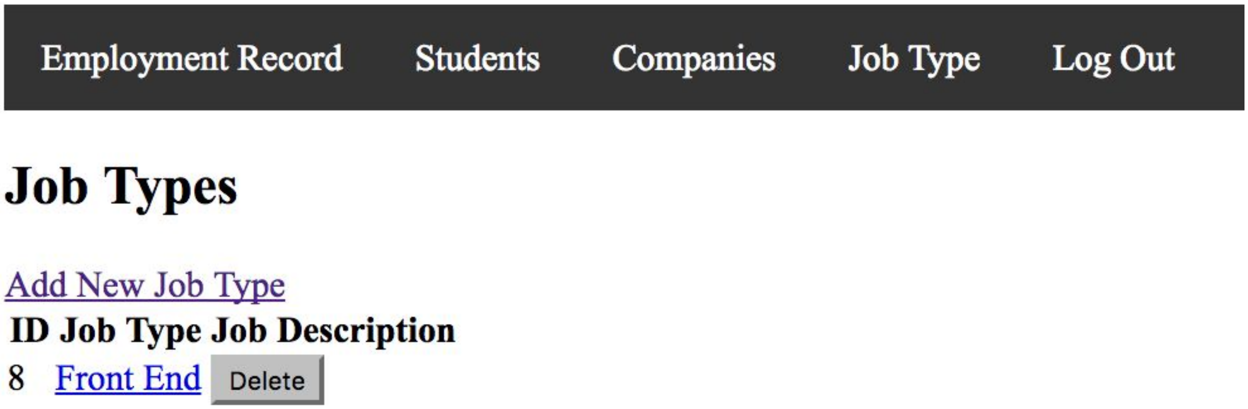


Student List

Add New Student

Student ID	Name	Gender	Cohort	Graduation Date	Company	
4	Craig Lynagh	Male	E14	2017-10-13	Alibaba	Delete
5	Charlie Chaplin	Male	G2	2017-04-10	CodeClan	Delete
6	Andrew Arran	Male	E14	2017-10-13		Delete
7	Alice Prince	Female	G1	2016-09-16		Delete
8	Stacy	Female	E14	2017-10-13		Delete
3	Chris Donnelly	Male	E14	2017-10-13	CodeClan	Delete
10	Ross Fisher	Male	G3	2017-10-21		Delete
11	Yan Ren	Female	E14	2017-10-13		Delete
2	Guy Bach	Male	E14	2017-10-13		Delete
12	Jack Ma	Male	E16	2018-01-31		Delete

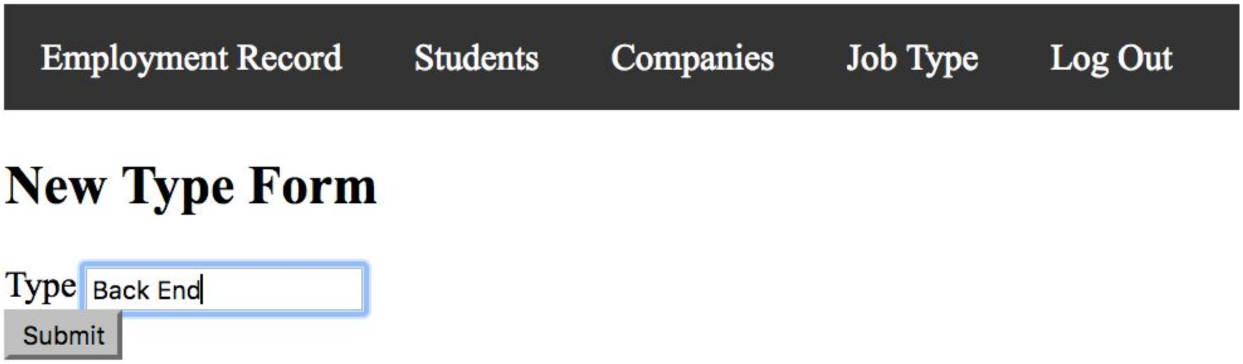
P14 Show an interaction with data persistence



Job Types

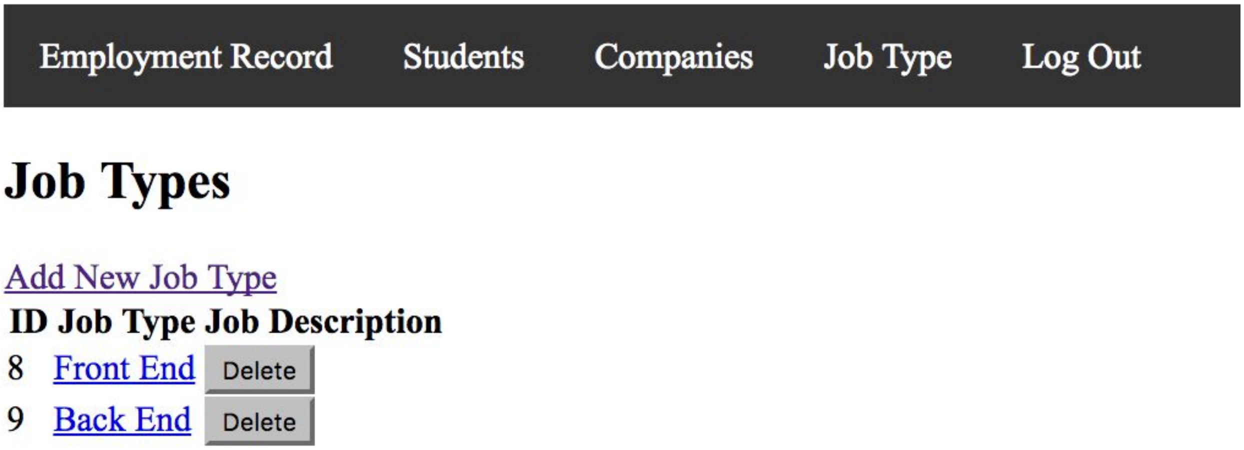
Add New Job Type

ID	Job Type	Job Description
8	Front End	Delete



New Type Form

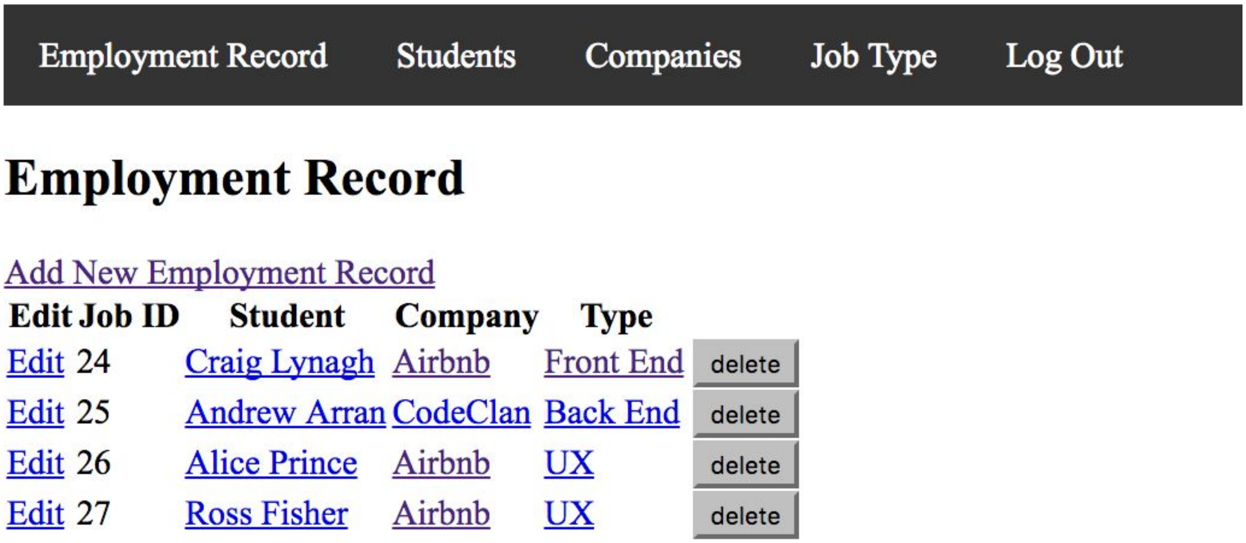
Type: Back End



Job Types

ID	Job Type	Job Description
8	Front End	Delete
9	Back End	Delete

P15 Show the correct output of results and feedback to user

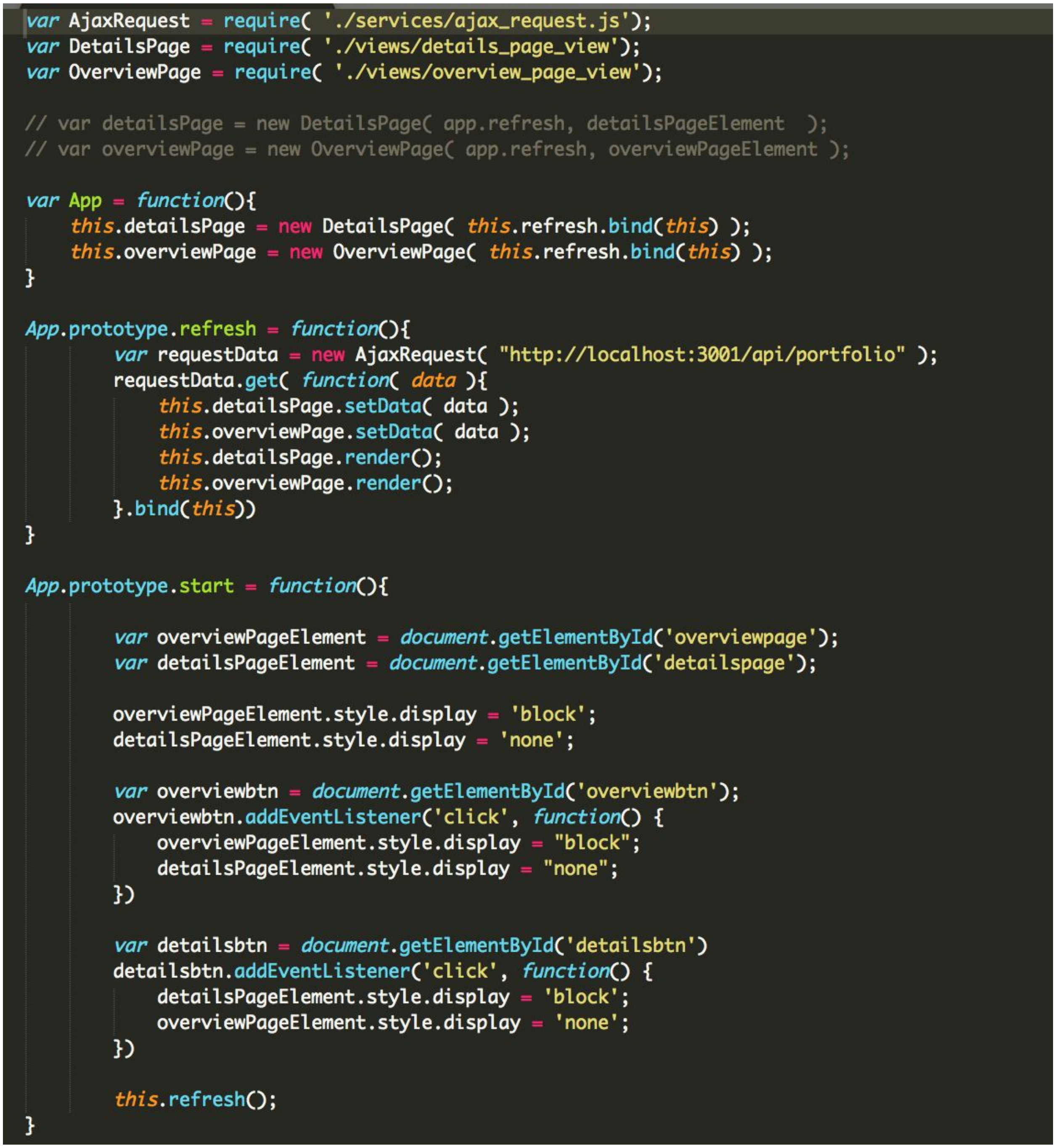


Employment Record

Add New Employment Record

Edit	Job ID	Student	Company	Type	
Edit	24	Craig Lynagh	Airbnb	Front End	delete
Edit	25	Andrew Arran	CodeClan	Back End	delete
Edit	26	Alice Prince	Airbnb	UX	delete
Edit	27	Ross Fisher	Airbnb	UX	delete

P16 API being used



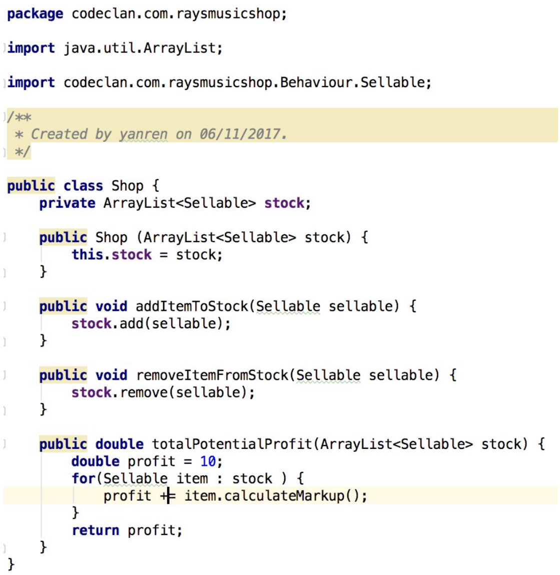
```
var AjaxRequest = require( './services/ajax_request.js');
var DetailsPage = require( './views/details_page_view');
var OverviewPage = require( './views/overview_page_view');

// var detailsPage = new DetailsPage( app.refresh, detailsPageElement  );
// var overviewPage = new OverviewPage( app.refresh, overviewPageElement );

var App = function(){
    this.detailsPage = new DetailsPage( this.refresh.bind(this) );
    this.overviewPage = new OverviewPage( this.refresh.bind(this) );
}

App.prototype.refresh = function(){
        var requestData = new AjaxRequest( "http://localhost:3001/api/portfolio" );
        requestData.get( function( data ){
            this.detailsPage.setData( data );
            this.overviewPage.setData( data );
            this.detailsPage.render();
            this.overviewPage.render();
        }.bind(this))
}

App.prototype.start = function(){

        var overviewPageElement = document.getElementById('overviewpage');
        var detailsPageElement = document.getElementById('detailspage');

        overviewPageElement.style.display = 'block';
        detailsPageElement.style.display = 'none';

        var overviewbtn = document.getElementById('overviewbtn');
        overviewbtn.addEventListener('click', function() {
            overviewPageElement.style.display = "block";
            detailsPageElement.style.display = "none";
        })

        var detailsbtn = document.getElementById('detailsbtn')
        detailsbtn.addEventListener('click', function() {
            detailsPageElement.style.display = 'block';
            overviewPageElement.style.display = 'none';
        })

        this.refresh();
}
```

P17 Bug tracking report

User can search company name	Fail	User can search company with SQL database	Pass
User can see the added up number of employees in a company	Fail	User can see add up number of employees with a new view	Pass
User can change company logos	Fail	User can upload documents and change logos	Pass
User can delete a company	Fail	User can delete a company with the delete button	Pass
User can see smaller size of of the student table	Fail	User can choose on the toggle button to see smaller size of table	Pass

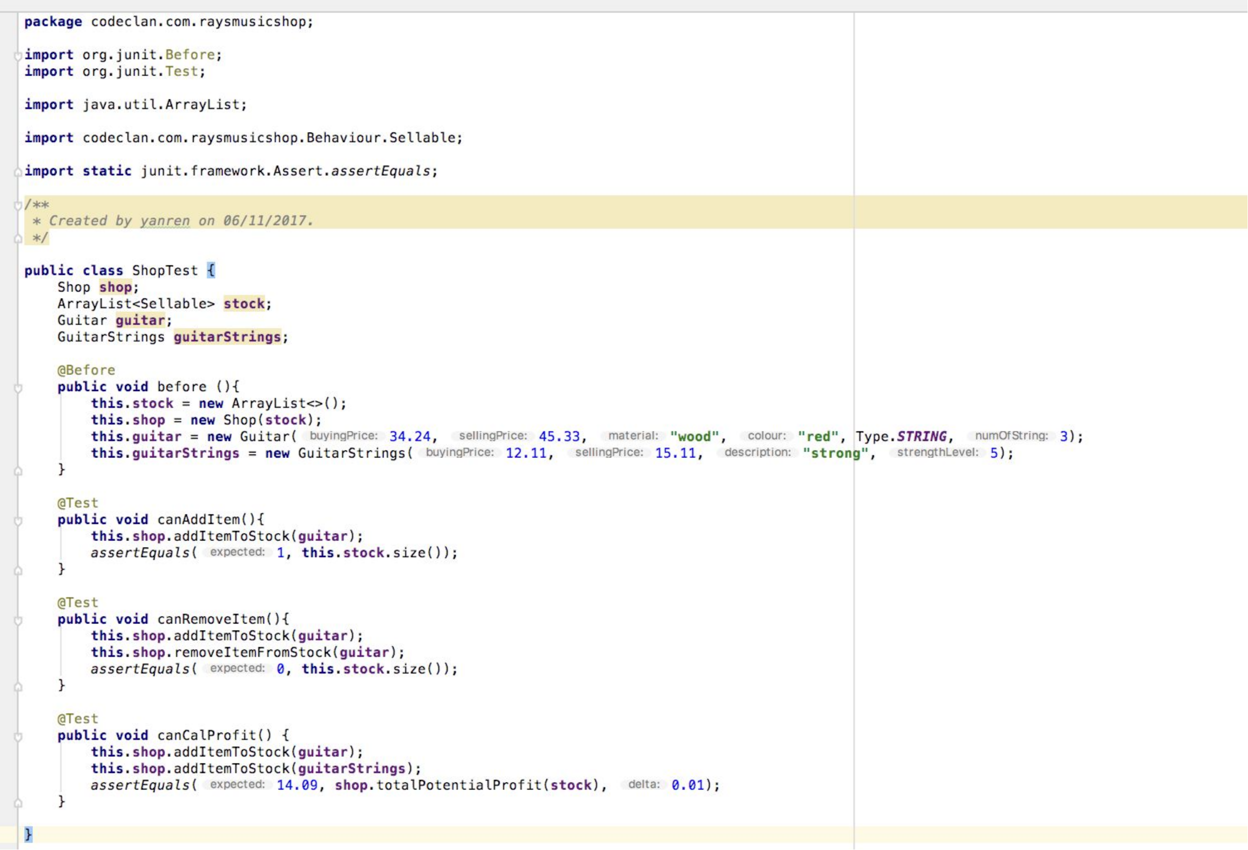
P18 Demonstration of testing

Example of the origin code



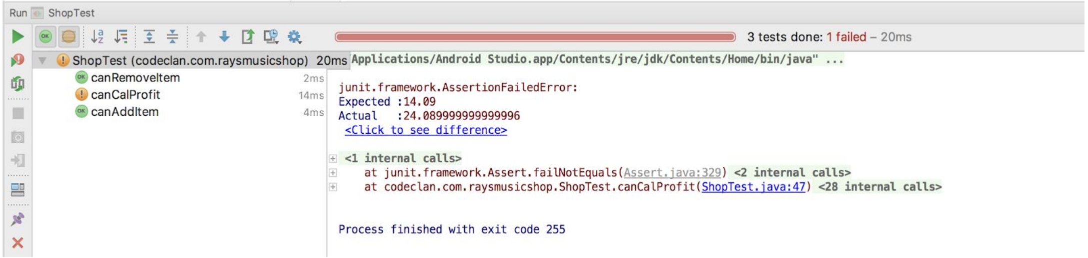
```
package codeclan.com.raysmusicshop;

import java.util.ArrayList;

import codeclan.com.raysmusicshop.Behaviour.Sellable;

/**
 * Created by yanren on 06/11/2017.
 */

public class Shop {
    private ArrayList<Sellable> stock;

    public Shop (ArrayList<Sellable> stock) {
        this.stock = stock;
    }

    public void addItemToStock(Sellable sellable) {
        stock.add(sellable);
    }

    public void removeItemFromStock(Sellable sellable) {
        stock.remove(sellable);
    }

    public double totalPotentialProfit(ArrayList<Sellable> stock) {
        double profit = 10;
        for(Sellable item : stock ) {
            profit += item.calculateMarkup();
        }
        return profit;
    }
}
```

Test file



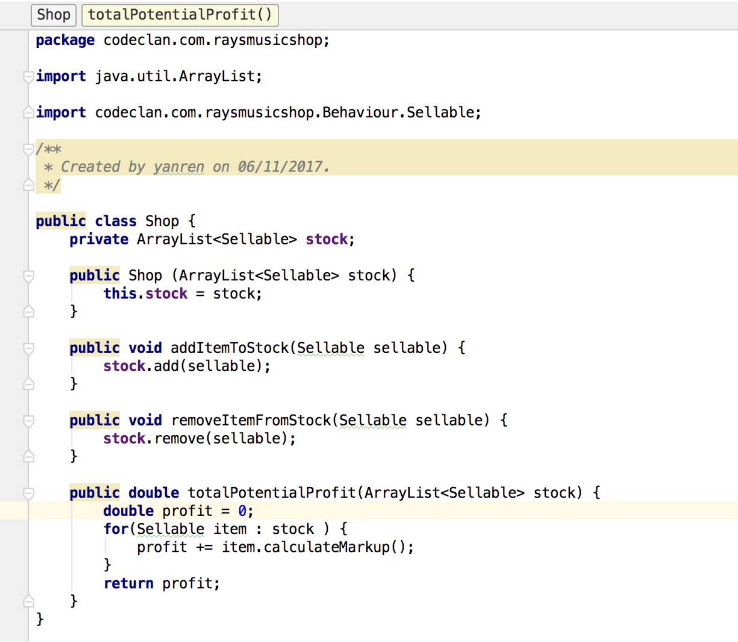
```java
package codeclan.com.raysmusicshop;

import org.junit.Before;
import org.junit.Test;

import java.util.ArrayList;

import codeclan.com.raysmusicshop.Behaviour.Sellable;

import static junit.framework.Assert.assertEquals;

/**
 * Created by yanren on 06/11/2017.
 */

public class ShopTest {
    Shop shop;
    ArrayList<Sellable> stock;
    Guitar guitar;
    GuitarStrings guitarStrings;

    @Before
    public void before (){
        this.stock = new ArrayList<>();
        this.shop = new Shop(stock);
        this.guitar = new Guitar( buyingPrice: 34.24,  sellingPrice: 45.33,  material: "wood",  colour: "red", Type.STRING,  numOfString: 3);
        this.guitarStrings = new GuitarStrings( buyingPrice: 12.11,  sellingPrice: 15.11,  description: "strong",  strengthLevel: 5);
    }

    @Test
    public void canAddItem(){
        this.shop.addItemToStock(guitar);
        assertEquals( expected: 1, this.stock.size());
    }

    @Test
    public void canRemoveItem(){
        this.shop.addItemToStock(guitar);
        this.shop.removeItemFromStock(guitar);
        assertEquals( expected: 0, this.stock.size());
    }

    @Test
    public void canCalProfit() {
        this.shop.addItemToStock(guitar);
        this.shop.addItemToStock(guitarStrings);
        assertEquals( expected: 14.09, shop.totalPotentialProfit(stock),  delta: 0.01);
    }

}
```

Test failed


Run
ShopTest
3 tests done: 1 failed – 20ms
ShopTest (codeclan.com.raysmusicshop) 20ms
canRemoveItem 2ms
canCalProfit 14ms
canAddItem 4ms

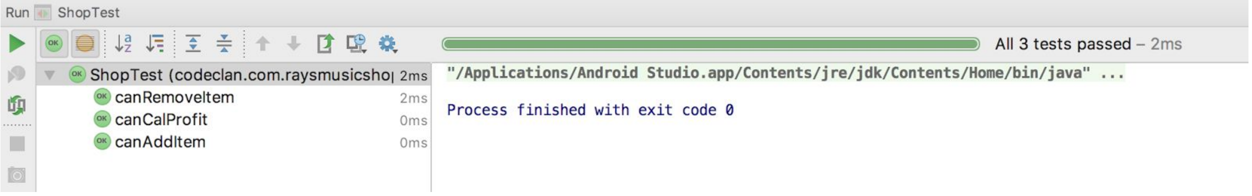
```
Applications/Android Studio.app/Contents/jre/jdk/Contents/Home/bin/java" ...

junit.framework.AssertionFailedError:
Expected :14.09
Actual   :24.089999999999996
 <Click to see difference>

 <1 internal calls>
    at junit.framework.Assert.failNotEquals(Assert.java:329) <2 internal calls>
    at codeclan.com.raysmusicshop.ShopTest.canCalProfit(ShopTest.java:47) <28 internal calls>


Process finished with exit code 255
```

Corrected code



```
package codeclan.com.raysmusicshop;

import java.util.ArrayList;

import codeclan.com.raysmusicshop.Behaviour.Sellable;

/**
 * Created by yanren on 06/11/2017.
 */

public class Shop {
    private ArrayList<Sellable> stock;

    public Shop (ArrayList<Sellable> stock) {
        this.stock = stock;
    }

    public void addItemToStock(Sellable sellable) {
        stock.add(sellable);
    }

    public void removeItemFromStock(Sellable sellable) {
        stock.remove(sellable);
    }

    public double totalPotentialProfit(ArrayList<Sellable> stock) {
        double profit = 0;
        for(Sellable item : stock ) {
            profit += item.calculateMarkup();
        }
        return profit;
    }
}
```

Test passed



```
All 3 tests passed – 2ms

ShopTest (codeclan.com.raysmusicsho|  2ms
    canRemoveItem                     2ms
    canCalProfit                      0ms
    canAddItem                        0ms
```

```
"/Applications/Android Studio.app/Contents/jre/jdk/Contents/Home/bin/java" ...

Process finished with exit code 0
```