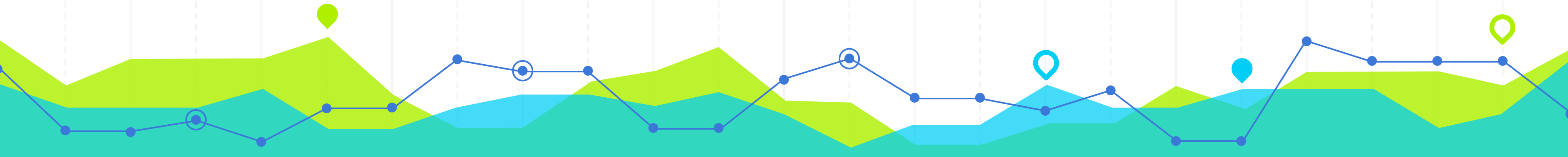


RACE ORGANIZER

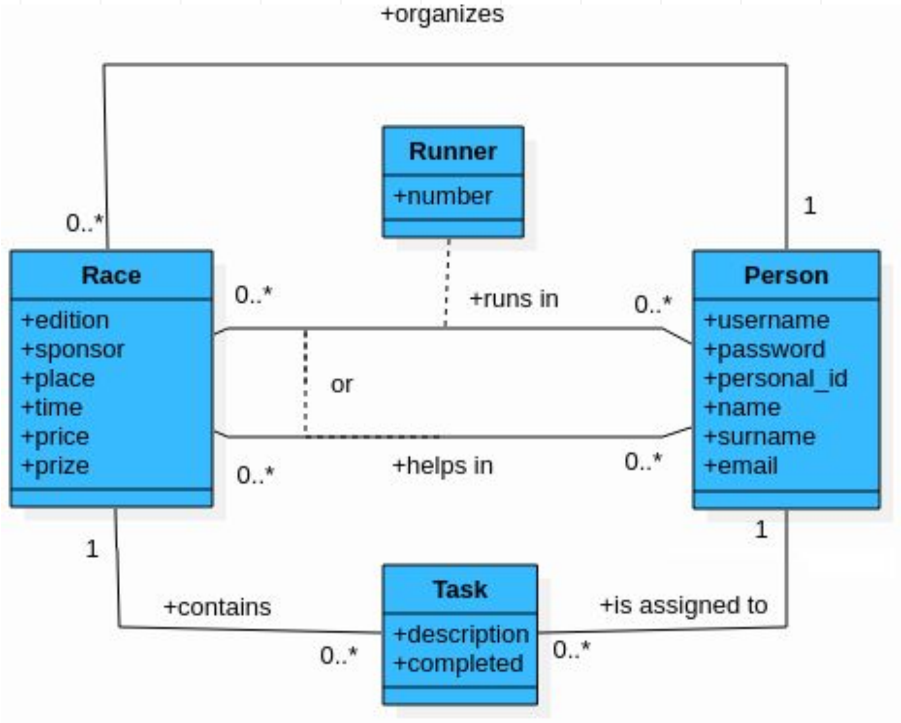
Group 8

Index

1. Purpose of the Application and its Architecture
2. Brief description of the process using SCRUM
3. Description of the use of YouTrack for Scrum and the integration with GitHub
4. Description of the use of Maven
5. Unit, Coverage and Performance Testing
6. Continuous Integration
7. Use of Doxygen and SLF4j
8. Use of Swagger
9. Events and Difficulties

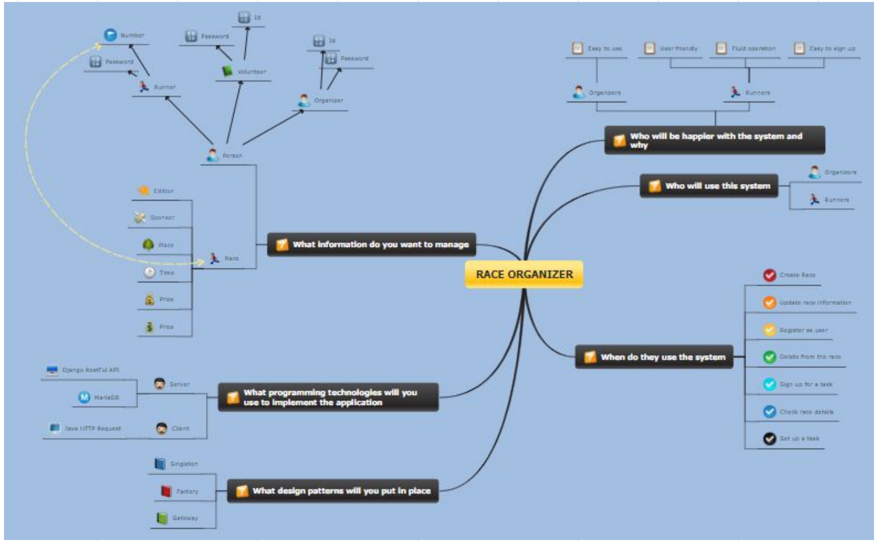
Purpose of the Application and its Architecture

- Sign up into a race
- Manage Race information
- Manage tasks
- REST client-server architecture
- Server on Django
- Client written in Java
- MariaDB

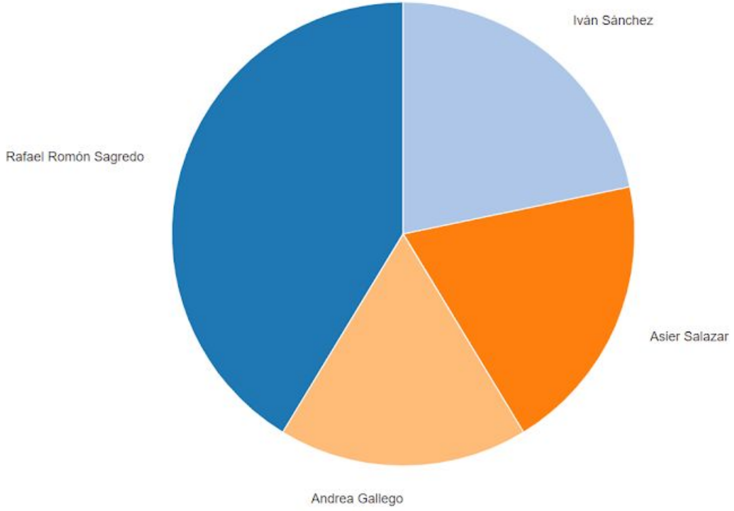


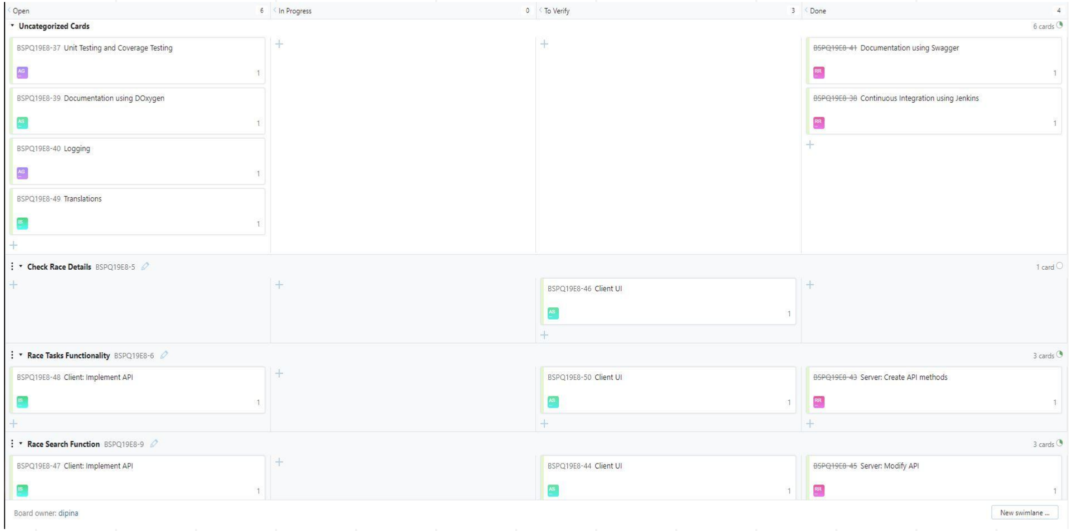
Brief description of the process using SCRUM

- Product owner
- Scrum master
- Product backlog
- Architecture
- Mindmap
- 3 sprints of 2 weeks - Meetings
  - Planning
  - Review
  - Retrospective



Description of the use of YouTrack for Scrum and the integration with GitHub



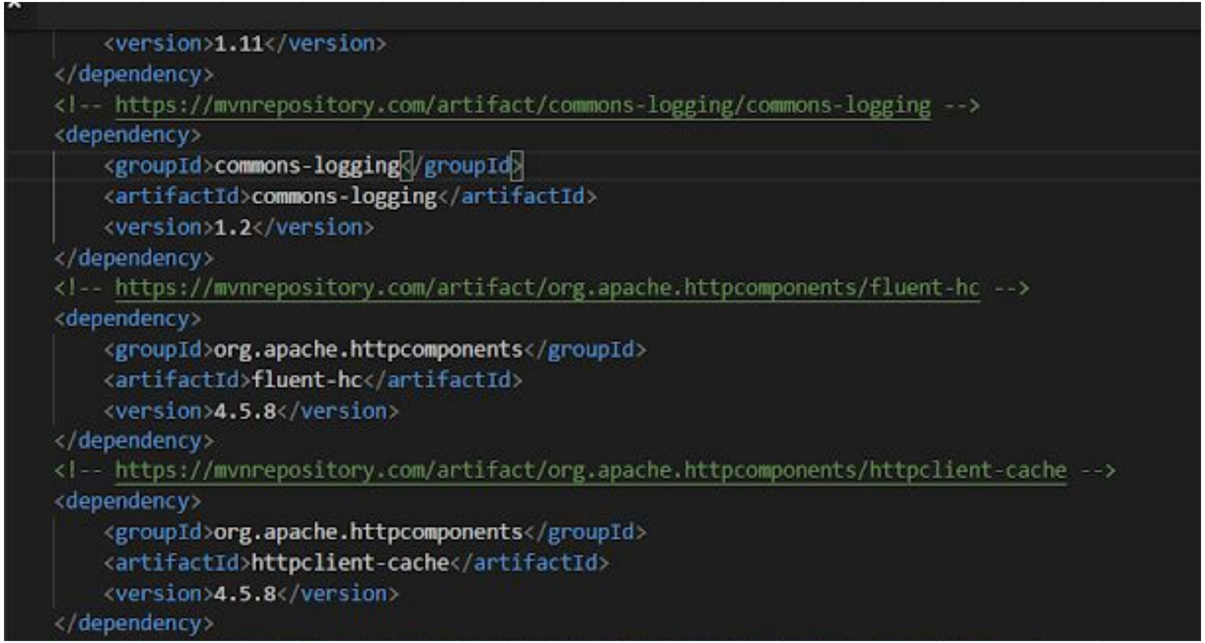


	Open 6	In Progress 0	To Verify 3	Done 4
Uncategorized Cards 6 cards	BSPQ19E8-37 Unit Testing and Coverage Testing AG 1 BSPQ19E8-39 Documentation using DOxygen AS 1 BSPQ19E8-40 Logging AG 1 BSPQ19E8-49 Translations IS 1			~~BSPQ19E8-41~~ Documentation using Swagger RR 1 ~~BSPQ19E8-38~~ Continuous Integration using Jenkins RR 1
Check Race Details BSPQ19E8-5 1 card			BSPQ19E8-46 Client UI AS 1	
Race Tasks Functionality BSPQ19E8-6 3 cards	BSPQ19E8-48 Client: Implement API IS 1		BSPQ19E8-50 Client UI AS 1	~~BSPQ19E8-43~~ Server: Create API methods RR 1
Race Search Function BSPQ19E8-9 3 cards	BSPQ19E8-47 Client: Implement API IS 1		BSPQ19E8-44 Client UI AS 1	~~BSPQ19E8-45~~ Server: Modify API RR 1

Board owner: dipina

Description of the use of Maven

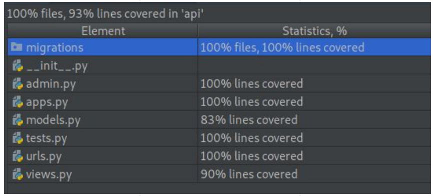
- Run Client
- Run Tests
- Dependencies/Plugins
  - HttpComponents
  - commons-logging
  - JUnit



```
    <version>1.11</version>
</dependency>
<!-- https://mvnrepository.com/artifact/commons-logging/commons-logging -->
<dependency>
    <groupId>commons-logging</groupId>
    <artifactId>commons-logging</artifactId>
    <version>1.2</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/fluent-hc -->
<dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>fluent-hc</artifactId>
    <version>4.5.8</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient-cache -->
<dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient-cache</artifactId>
    <version>4.5.8</version>
</dependency>
```

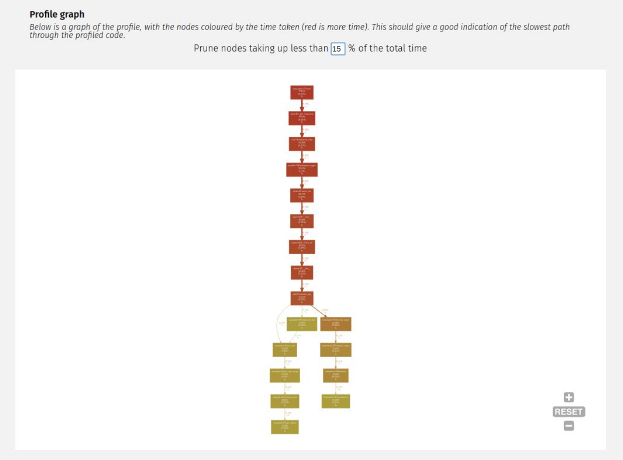
Unit, Coverage and Performance Testing

- Server Test → 93% Coverage
- Performance Testing using Locust
- Profiling silk



100% files, 93% lines covered in 'api'

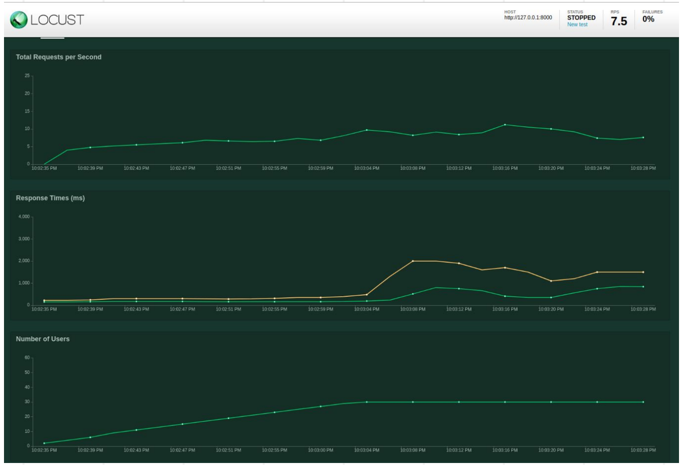
Element	Statistics, %
migrations	100% files, 100% lines covered
__init__.py	
admin.py	100% lines covered
apps.py	100% lines covered
models.py	83% lines covered
tests.py	100% lines covered
urls.py	100% lines covered
views.py	90% lines covered



Profile graph

Below is a graph of the profile, with the nodes coloured by the time taken (red is more time). This should give a good indication of the slowest path through the profiled code.

Prune nodes taking up less than 15 % of the total time



LOCUST

HOST http://127.0.0.1:8000

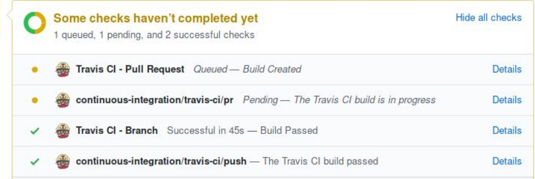
STATUS STOPPED

New test

RPS 7.5

FAILURES 0%

Continuous Integration

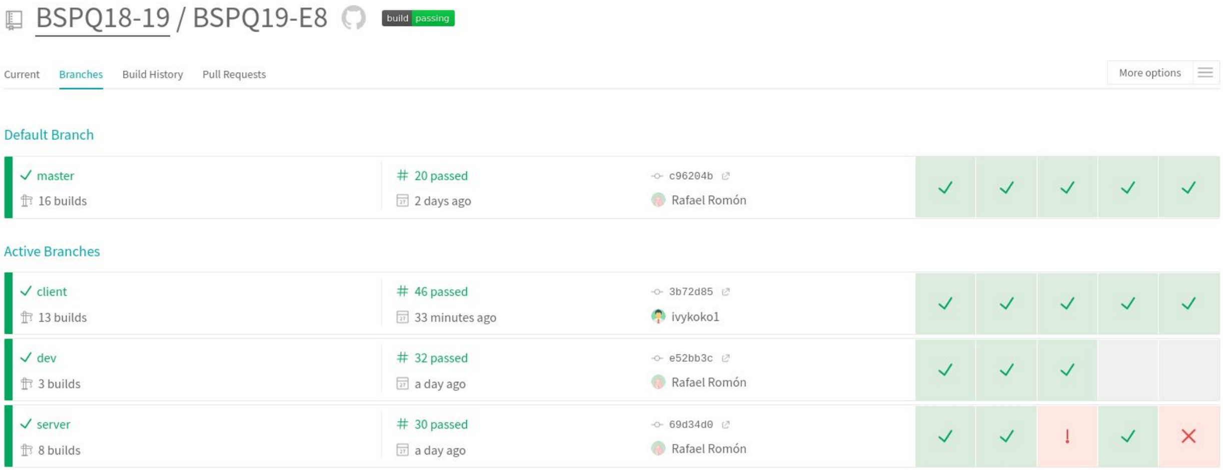


Some checks haven't completed yet

1 queued, 1 pending, and 2 successful checks

Hide all checks

●	**Travis CI - Pull Request** *Queued — Build Created*	Details
●	**continuous-integration/travis-ci/pr** *Pending — The Travis CI build is in progress*	Details
✓	**Travis CI - Branch** Successful in 45s — Build Passed	Details
✓	**continuous-integration/travis-ci/push** — The Travis CI build passed	Details



BSPQ18-19 / BSPQ19-E8 build passing

Current Branches Build History Pull Requests More options

Default Branch

Branch	Builds	Passed	When	Commit	Author					
✓ master	16 builds	20 passed	2 days ago	c96204b	Rafael Romón	✓	✓	✓	✓	✓

Active Branches

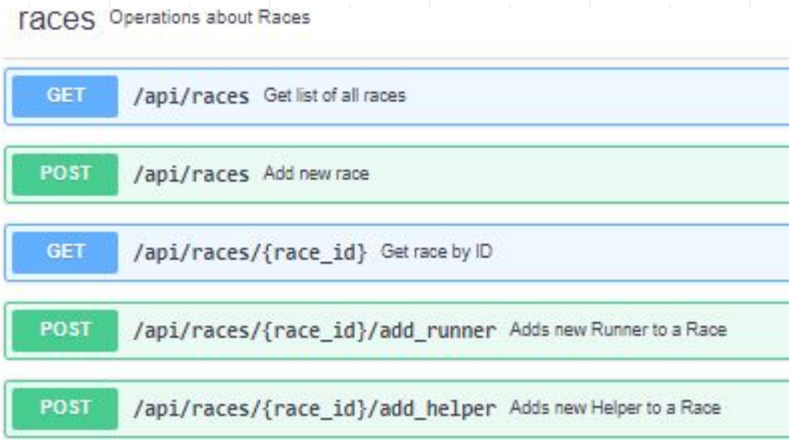
Branch	Builds	Passed	When	Commit	Author					
✓ client	13 builds	46 passed	33 minutes ago	3b72d85	ivykoko1	✓	✓	✓	✓	✓
✓ dev	3 builds	32 passed	a day ago	e52bb3c	Rafael Romón	✓	✓	✓		
✓ server	8 builds	30 passed	a day ago	69d34d0	Rafael Romón	✓	✓	!	✓	✕

Use of Doxygen and SLF4j

- Doxygen is an automated multi-language doc generator.
- It allows to export docs to PDF, LaTeX or HTML.
- Simple Logging Facade for Java (SLF4J) serves as a simple facade or abstraction for various logging frameworks such as Log4j, which we have used.

Use of Swagger

- Swagger makes it easy to visualize RESTful APIs and their responses.
- Documentation for the API follows the OpenAPI standard



races Operations about Races

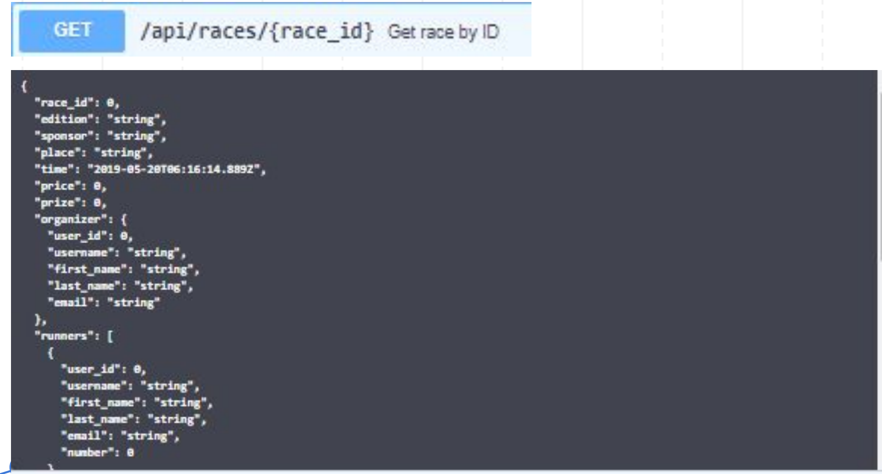
GET /api/races Get list of all races

POST /api/races Add new race

GET /api/races/{race_id} Get race by ID

POST /api/races/{race_id}/add_runner Adds new Runner to a Race

POST /api/races/{race_id}/add_helper Adds new Helper to a Race



GET /api/races/{race_id} Get race by ID

```
{
  "race_id": 0,
  "edition": "string",
  "sponsor": "string",
  "place": "string",
  "time": "2019-05-20T06:16:14.889Z",
  "price": 0,
  "prize": 0,
  "organizer": {
    "user_id": 0,
    "username": "string",
    "first_name": "string",
    "last_name": "string",
    "email": "string"
  },
  "runners": [
    {
      "user_id": 0,
      "username": "string",
      "first_name": "string",
      "last_name": "string",
      "email": "string",
      "number": 0
```

Events and Difficulties

- Main difficulty: integration and learning new technologies.
- Lack of communication makes integration between two different platforms more difficult.
- Server DB creation difficulties using MySQL instead of MariaDB.
- Adapting to use a project management tool, like YouTrack.
- Time management.