

# SAPACA

Pascal Treptow, Sascha Kühne, Carolina Mehret

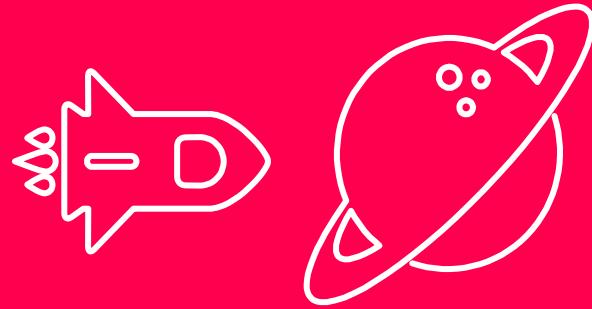


# sapaca

## Agenda

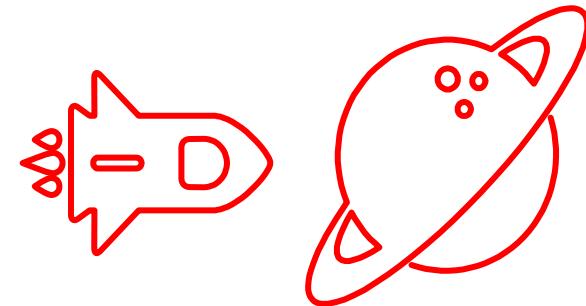
- Vision
- Rollen
- SRS - Software Requirement Specification
- Projectmanagement
- Architecture
- Quality Assurance
- Demo

# VISION



# sapaca

## Vision



- Applikation, die **Face Detection** und **Klassifizierungen** realisiert
- Applikation soll über eine Webseite erreichbar und bedienbar sein
- Speichern und Auswerten von Daten

# ROLLEN

# sapaca

## Rollen



### Pascal Treptow

- Software Architekt
- Designer
- Test Manager
- Implementierer



### Sascha Kühne

- Software Architekt
- Test Manager
- Implementierer
- Konfigurations- manager

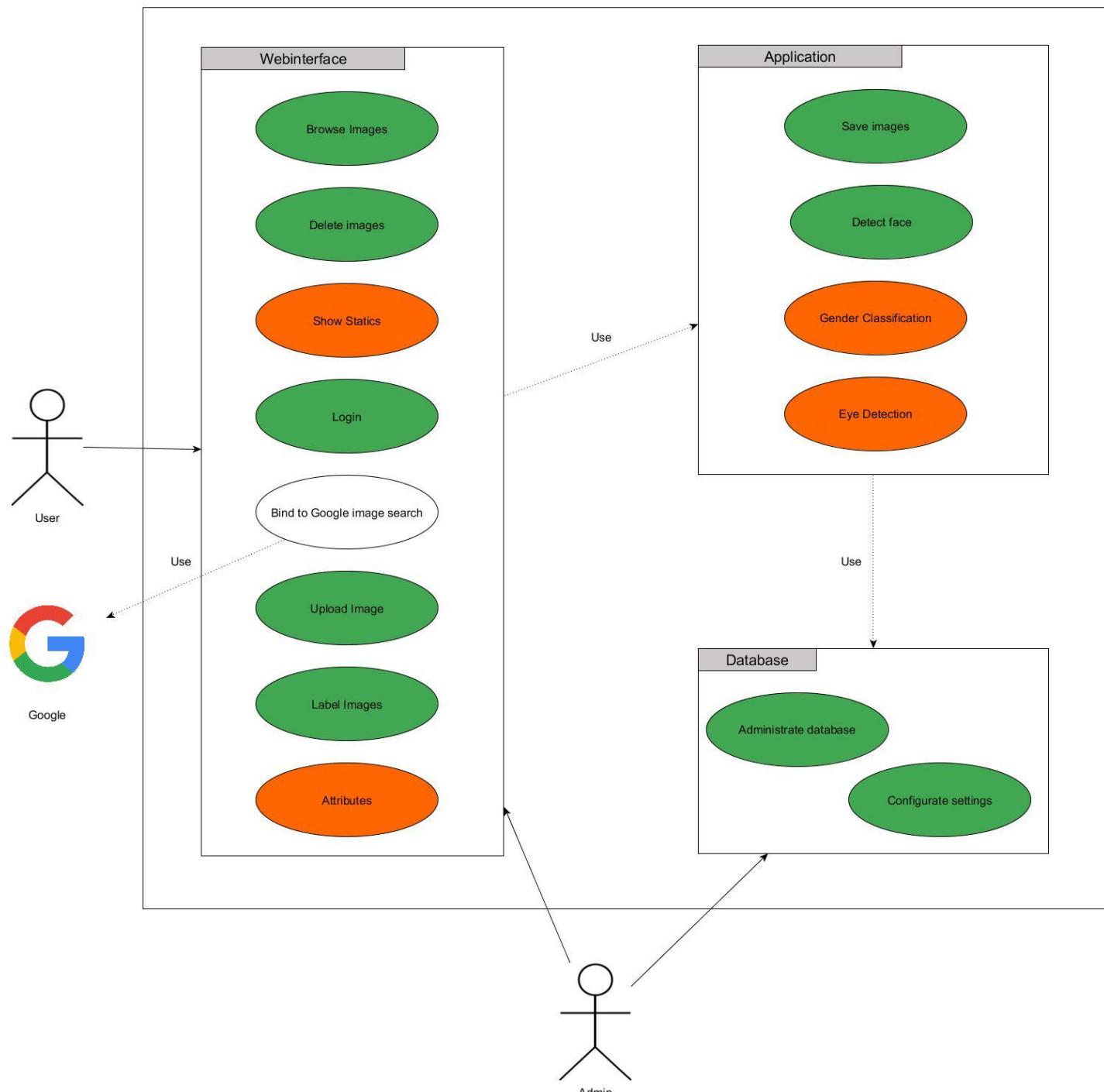


### Carolina Mehret

- Projektmanager
- System Analyst
- Implementierer

# SRS

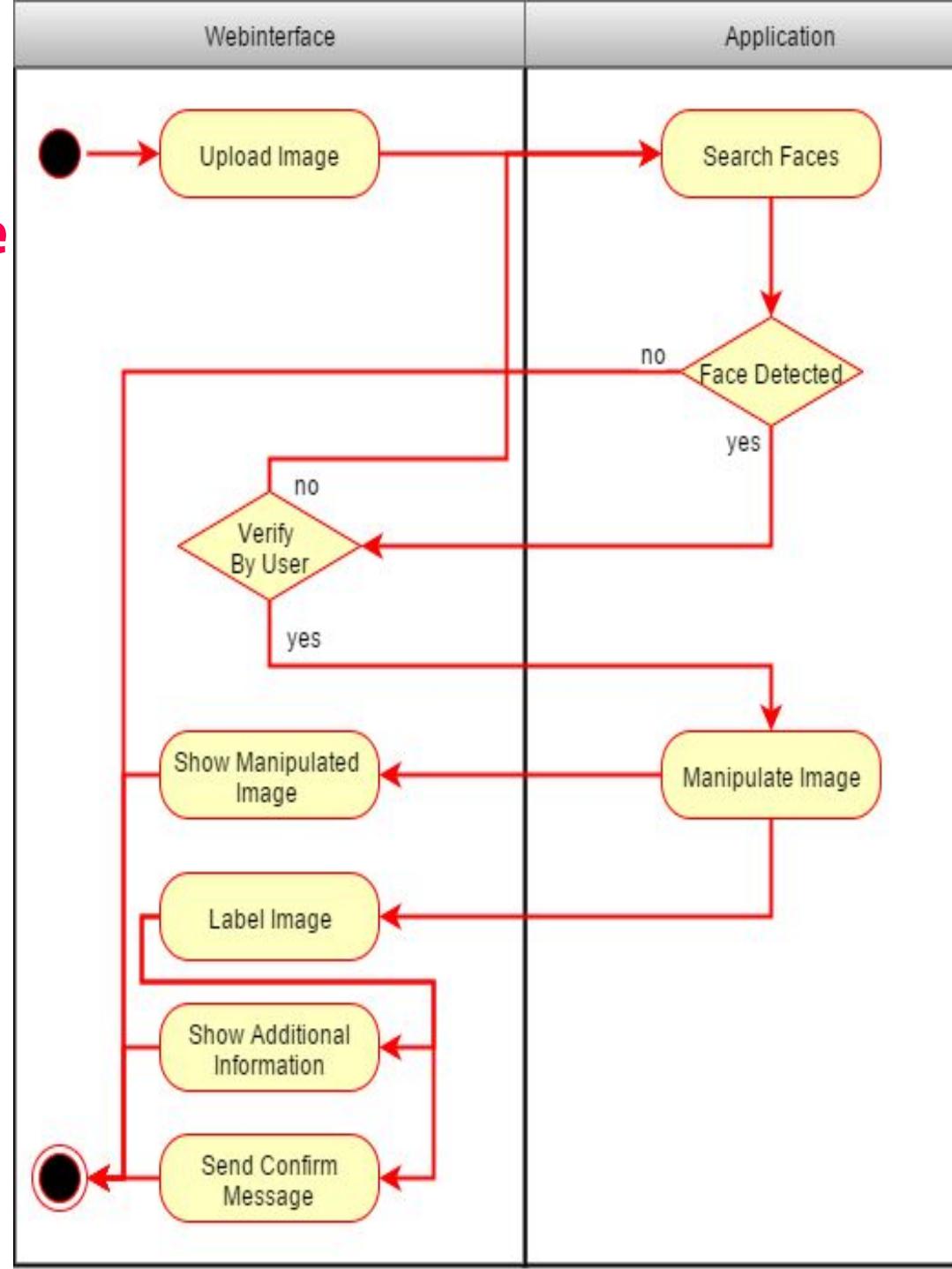
Software Requirement Specification



**sapaca**

## Use Case: Detect Face

- Verwendung von Activity Diagram



**sapaca**

## **Non-Functional Requirements**

- Response Time: Die Applikation muss jede Anfrage des Benutzers innerhalb von 5 Sekunden durchführen
- Bedienbarkeit: Durch die eine intuitive Benutzeroberfläche der Applikation soll kein Lernaufwand für den Benutzer anfallen
- Plattformunabhängigkeit

**sapaca**

## **SRS - Software Requirement Specification**

- Beinhaltet Informationen zur Applikation allgemein und zur benötigten Hardware
  - Definiert, wie die Applikation mit dem System und dem User interagiert
  - Stellt sicher, dass die Requirements vom Entwicklungsteam korrekt verstanden wurden
- 
- <https://github.com/sapacaFaceRecognition/Documentation/wiki/Software-Requirements-Specification>

# PROJECT MANAGEMENT

# **sapaca**

## **SCRUM**

- Eine Methode zur agilen Softwareentwicklung
  - Basiert auf Sprints
- 
- Wöchentliches Planen
  - Aufteilung der Arbeit
  - Fortschritt gut sichtbar

# sapaca

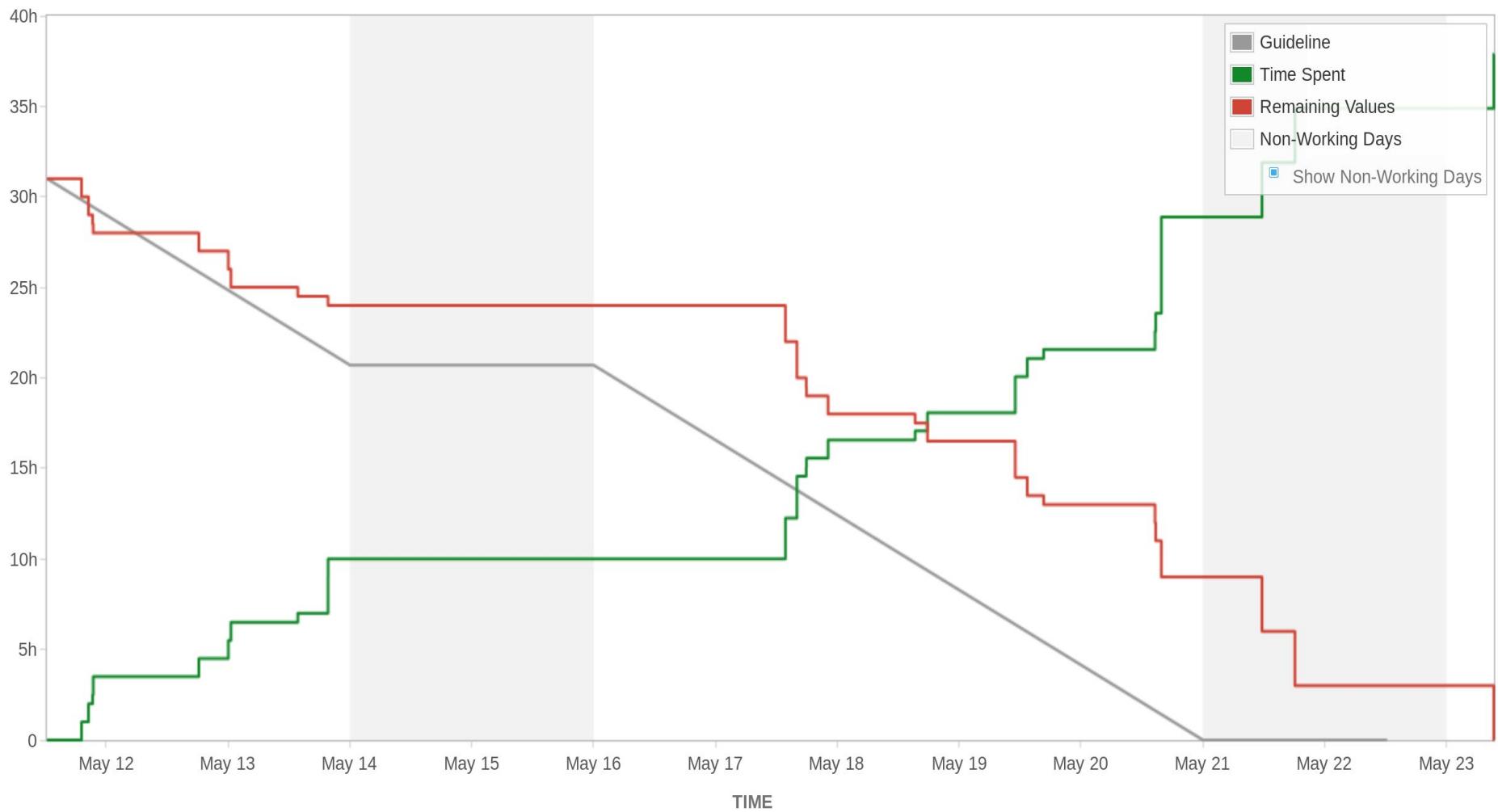
## JIRA

Bild von Backlog, erklären wie JIRA SCRUM unterstützt  
Planning mit JIRA anhand vom Bild erklären

Abgeschlossene Vorgänge						Im Vorgangsnavigator anzeigen
Schlüssel	Zusammenfassung	Vorgangstyp	Priorität	Status	Ursprüngliche Zeitschätzung (3d 7h)	
SAVI-79	Gender Classification: Train model	Story	↑ Medium	FERTIG		4h
SAVI-80	Eye Detection: Complete image instead of cropped ones	Bug	↑ Medium	FERTIG		1h
SAVI-81	Update Class Diagrams with MVC view	Story	↑ Medium	FERTIG		1h
SAVI-82	RUP: Add Deployment view	Story	↑ Medium	FERTIG		30m
SAVI-83	Write Tests for Detector	Story	↑ Medium	FERTIG		2h
SAVI-84	FileUploader bauen	Story	↑ Medium	FERTIG		3h
SAVI-85	Build ImageUploader on Cloudinary (UC: Gender Classification)	Story	↑ Medium	FERTIG		2h
SAVI-86	Test Plan	Story	↑ Medium	FERTIG		3h
SAVI-87	Stress Test	Story	↑ Medium	FERTIG		3h
SAVI-88	Blogeintrag	Story	↑ Medium	FERTIG		30m
SAVI-89	Attributes implementieren GUI	Story	↑ Medium	FERTIG		2h
SAVI-90	Attributes implementieren	Story	↑ Medium	FERTIG		2h
SAVI-91	Eye Detection anbindung	Story	↑ Medium	FERTIG		2h
SAVI-92	Statistics implementieren	Story	↑ Medium	FERTIG		2h
SAVI-93	Bind to Google Search7	Story	↑ Medium	FERTIG		3h

# sapaca

## JIRA



# sapaca

## Risk Management

Risk Name	Risk Description	Risk Probability of Occurrence	Risk Impact (1-10)	Risk Factor	Risk Mitigation	Person in Charge
Klausurenphase	Zeitplanung des Projekts wird negativ durch Klausurenphase beeinflusst.	75%	10	7.5	- Frühzeitige Planung von Aufgaben - Detailierte Planung - Weniger Requirements - Weniger prokrastinieren, mehr lernen	C
Unrealistische Zeitschätzung	Projekt kommt in Verzug durch unrealistische Zeitschätzung	50%	8	4	- Aufgaben in kleine Tasks zerlegen, einfacher abzuschätzen - Im Vorfeld über Aufgaben informieren, um diese besser abschätzen zu können- Zeit zum Testen planen	S

- <https://github.com/sapacaFaceRecognition/Documentation/wiki/Risk-Management>

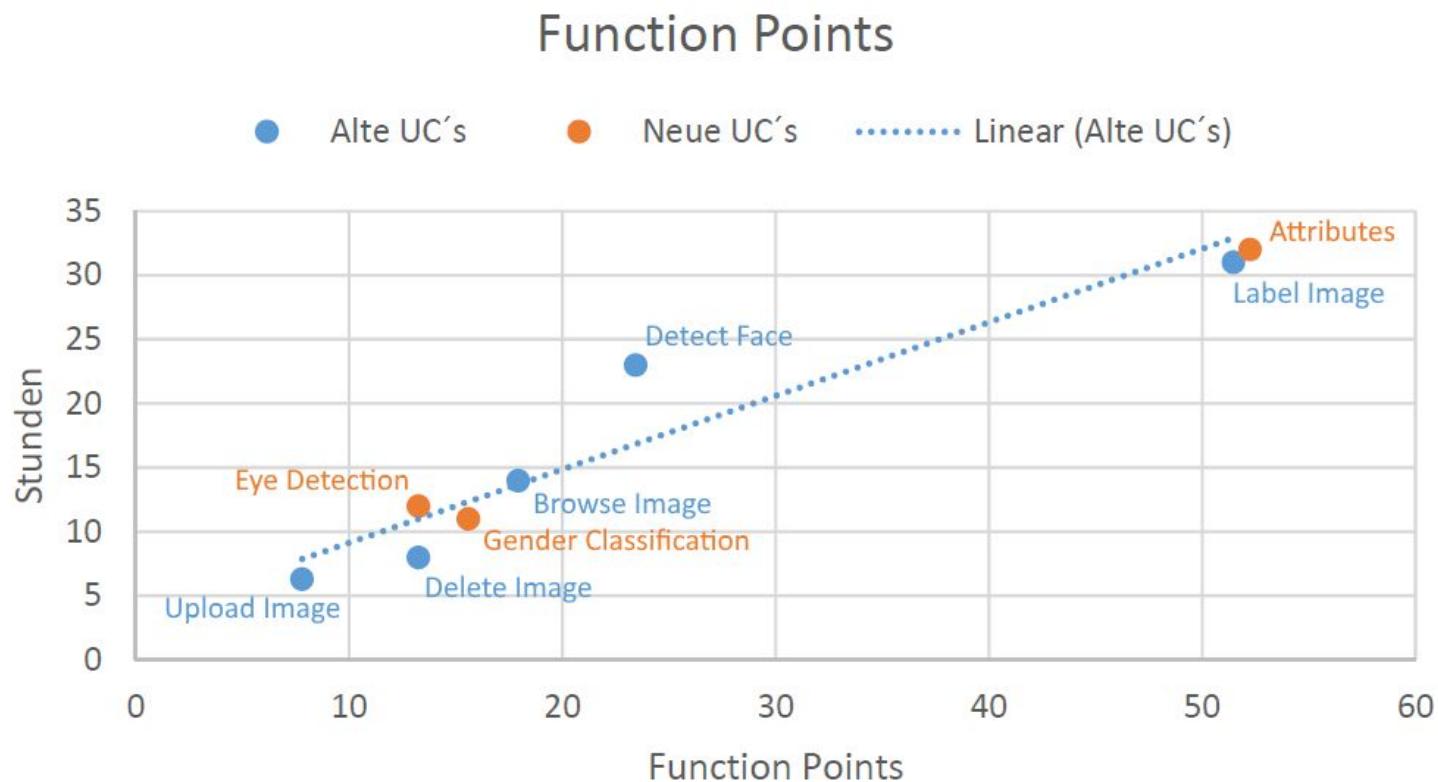
# sapaca

## Function Points

- Berechnung von Komplexität eines Use Cases
- Use Cases bekommen eine FP Zahl
- Verknüpft mit der gebrauchten Zeit pro Use Case
- Neue Use Cases können einfach geschätzt werden

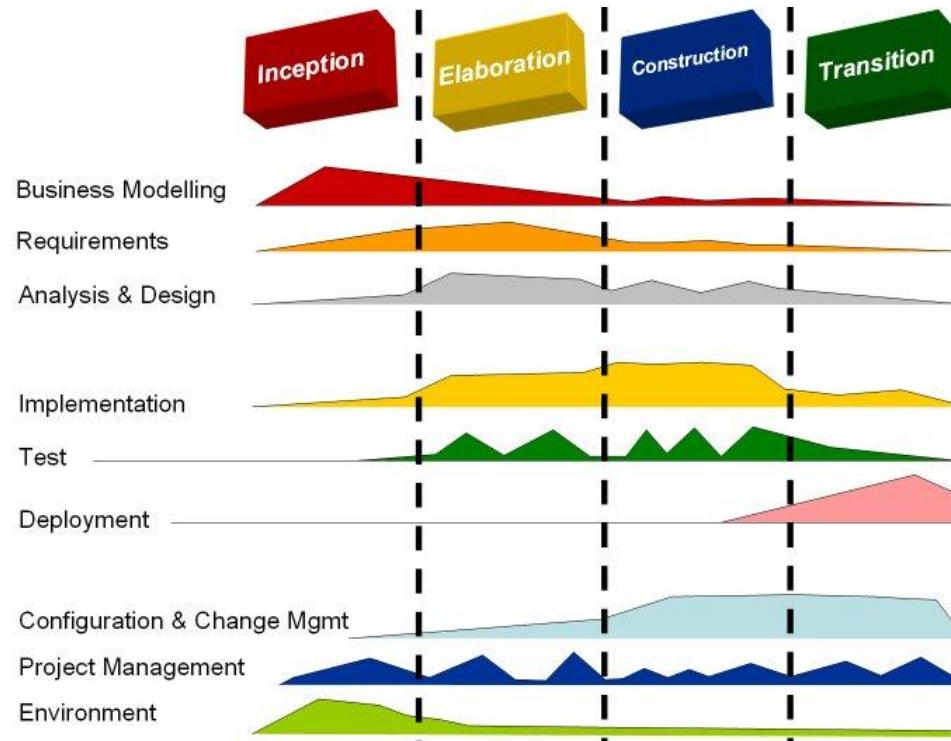
# sapaca

## Function Points

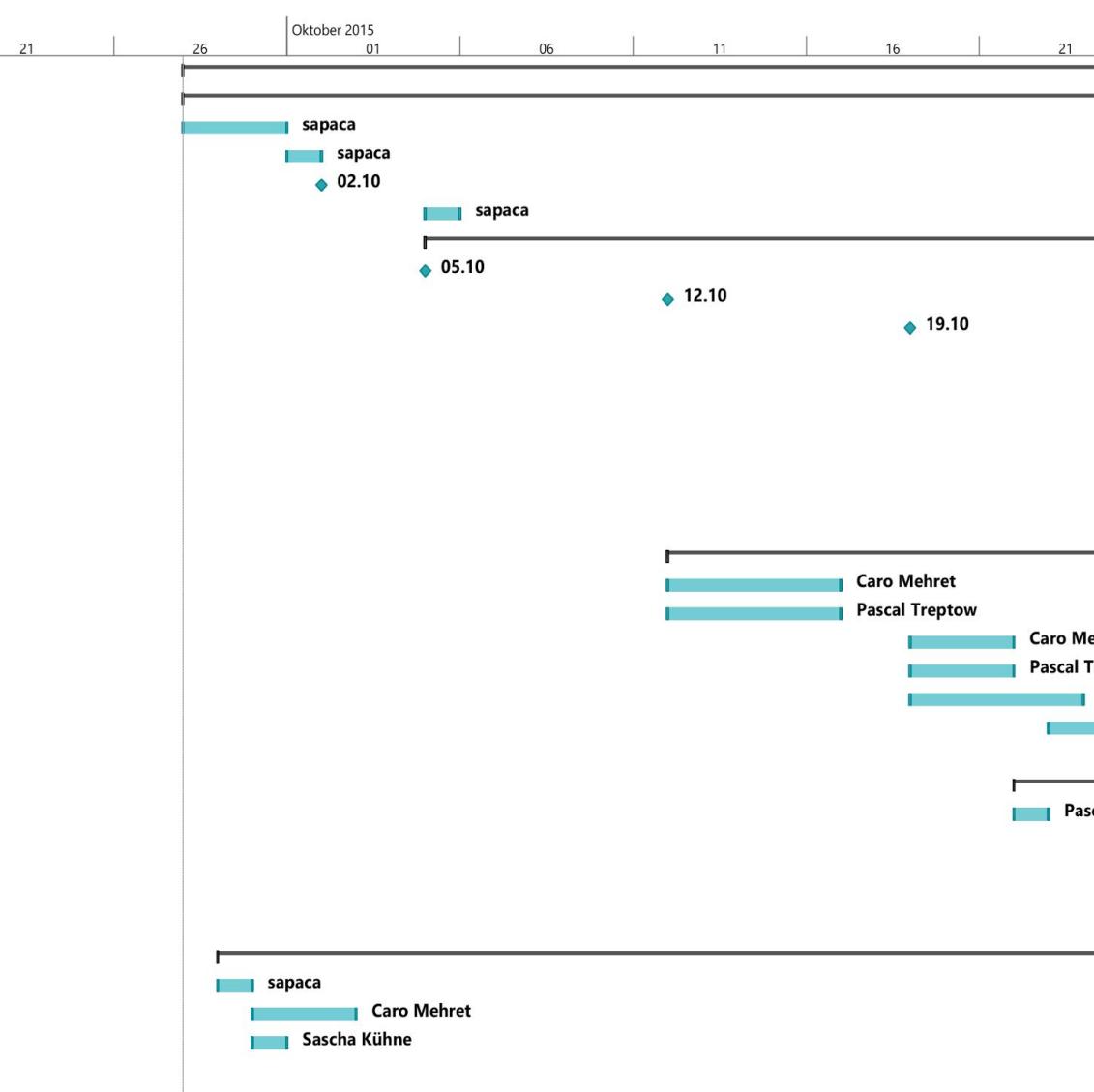


sapaca

# RUP - Rational Unified Process



<b>INCEPTION</b>			
<b>Project Management</b>	<b>46 Tage</b>	<b>Mon 28.09.15</b>	<b>Mon 30.11.15</b>
Conceive new Project	3 Tage	Mon 28.09.15	Mit 30.09.15
Project Scope and Risks	1 Tag	Don 01.10.15	Don 01.10.15
Project acceptance	0 Tage	Fre 02.10.15	Fre 02.10.15
Refine Role and Responsibilities	1 Tag	Mon 05.10.15	Mon 05.10.15
<b>Homeworks</b>	<b>40 Tage</b>	<b>Mon 05.10.15</b>	<b>Mon 30.11.15</b>
Homework 1	0 Tage	Mon 05.10.15	Mon 05.10.15
Homework 2	0 Tage	Mon 12.10.15	Mon 12.10.15
Homework 3	0 Tage	Mon 19.10.15	Mon 19.10.15
Homework 4	0 Tage	Mon 26.10.15	Mon 26.10.15
Homework 5	0 Tage	Mon 02.11.15	Mon 02.11.15
Homework 6	0 Tage	Mon 09.11.15	Mon 09.11.15
Homework 7	0 Tage	Mon 16.11.15	Mon 16.11.15
Homework 8	0 Tage	Mon 23.11.15	Mon 23.11.15
Homework 9	0 Tage	Mon 30.11.15	Mon 30.11.15
<b>Business Modelling</b>			
<b>Requirements</b>	<b>19 Tage</b>	<b>Mon 12.10.15</b>	<b>Don 05.11.15</b>
SRS	5 Tage	Mon 12.10.15	Fre 16.10.15
Overall UCD	5 Tage	Mon 12.10.15	Fre 16.10.15
Use Case Detect Face	3 Tage	Mon 19.10.15	Mit 21.10.15
Use Case Label Image	3 Tage	Mon 19.10.15	Mit 21.10.15
Use Case Browse Image	5 Tage	Mon 19.10.15	Fre 23.10.15
Use Case Upload Image	5 Tage	Fre 23.10.15	Don 29.10.15
Use Case Delete Image	5 Tage	Fre 30.10.15	Don 05.11.15
<b>Analysis und Design</b>	<b>17 Tage</b>	<b>Don 22.10.15</b>	<b>Fre 13.11.15</b>
Mockups	1 Tag	Don 22.10.15	Don 22.10.15
Class Diagramm	3 Tage	Mon 02.11.15	Mit 04.11.15
Architecture	10 Tage	Mon 02.11.15	Fre 13.11.15
SAD	3 Tage	Mon 09.11.15	Mit 11.11.15
<b>Test</b>			
<b>Environment</b>	<b>45 Tage</b>	<b>Die 29.09.15</b>	<b>Mon 30.11.15</b>
Set up GitHub	1 Tag	Die 29.09.15	Die 29.09.15
Set up OpenCV and Gradle	3 Tage	Mit 30.09.15	Fre 02.10.15
Set up Spring MVC	1 Tag	Mit 30.09.15	Mit 30.09.15
Set up JIRA	2 Tage	Don 19.11.15	Fre 20.11.15
Set in Gantt Chart	5 Tage	Die 24.11.15	Mon 30.11.15



[https://github.com/sapacaFaceRecognition/Documentation/blob/master/MS%20Project/FaceRecognition\\_Projektplanung.pdf](https://github.com/sapacaFaceRecognition/Documentation/blob/master/MS%20Project/FaceRecognition_Projektplanung.pdf)

# ARCHITECTURE

# sapaca

## Technologien

- **IDE:** IntelliJ, Eclipse
- **Build Management:** Gradle
- **Sprachen:** Java, C++, Python
- **Libraries:** OpenCV/JavaCV
- **Datenbank:** MySQL, Spring Data, Hibernate
- **Versionskontrolle:** GitHub
- **Sonstiges:** Spring Boot, Spring Web MVC, Thymeleaf

# sapaca

## JIRA Integration

The screenshot shows a JIRA Task List window with a dark theme. At the top, there is a header bar with a green circular icon on the left, followed by the text "Task List" and a close button "X". Below the header is a search bar containing the placeholder text "Find" and a dropdown menu with options "All" and "Activate...". The main area is a list of JIRA issues, each represented by a small icon, a priority indicator (e.g., blue dot), and a ticket number followed by a brief description. The issues are listed vertically:

- SAVI-44: SonarQube
- SAVI-47: Gender Classification in Spring implementieren + zusätzliches Datenbankattribut
- SAVI-48: Label Image, Feld für Geschlecht einfügen
- SAVI-49: Restlichen Seiten aufbauen
- SAVI-50: Favicon Fehler
- SAVI-51: Webseiten und Mockup überarbeiten
- SAVI-52: Unit-Test MVC
- SAVI-53: Gender Classification
- SAVI-54: TDD
- SAVI-55: Google Custom Search API implementieren + Image-Search testen
- SAVI-56: TDD Video aufnehmen
- SAVI-57: Unit tests erweitern + Testdatenbank einrichten/verwenden
- SAVI-58: Image Search Alternative finden
- SAVI-59: Fowler Refactor Care
- SAVI-60: Fowler Refactor Sascha
- SAVI-61: Fowler Refactor Pascal
- SAVI-62: Gender Classification: Fishermans Algorithmus implementieren
- SAVI-63: Eye Detection: Dokumentation
- SAVI-64: Eye Detection: Implementierung Haarcascading Algorithmus
- SAVI-65: Continuous Integration
- SAVI-66: FP neu berechnen

# sapaca

## Database

The image shows a screenshot of a database management system interface. It displays two tables side-by-side.

**faces Table:**

- id BIGINT(20)
- age INT(11)
- image LONGBLOB
- first\_name VARCHAR(255)
- gender INT(11)
- last\_name VARCHAR(255)
- location VARCHAR(255)
- nationality VARCHAR(255)

**Indexes:** [button]

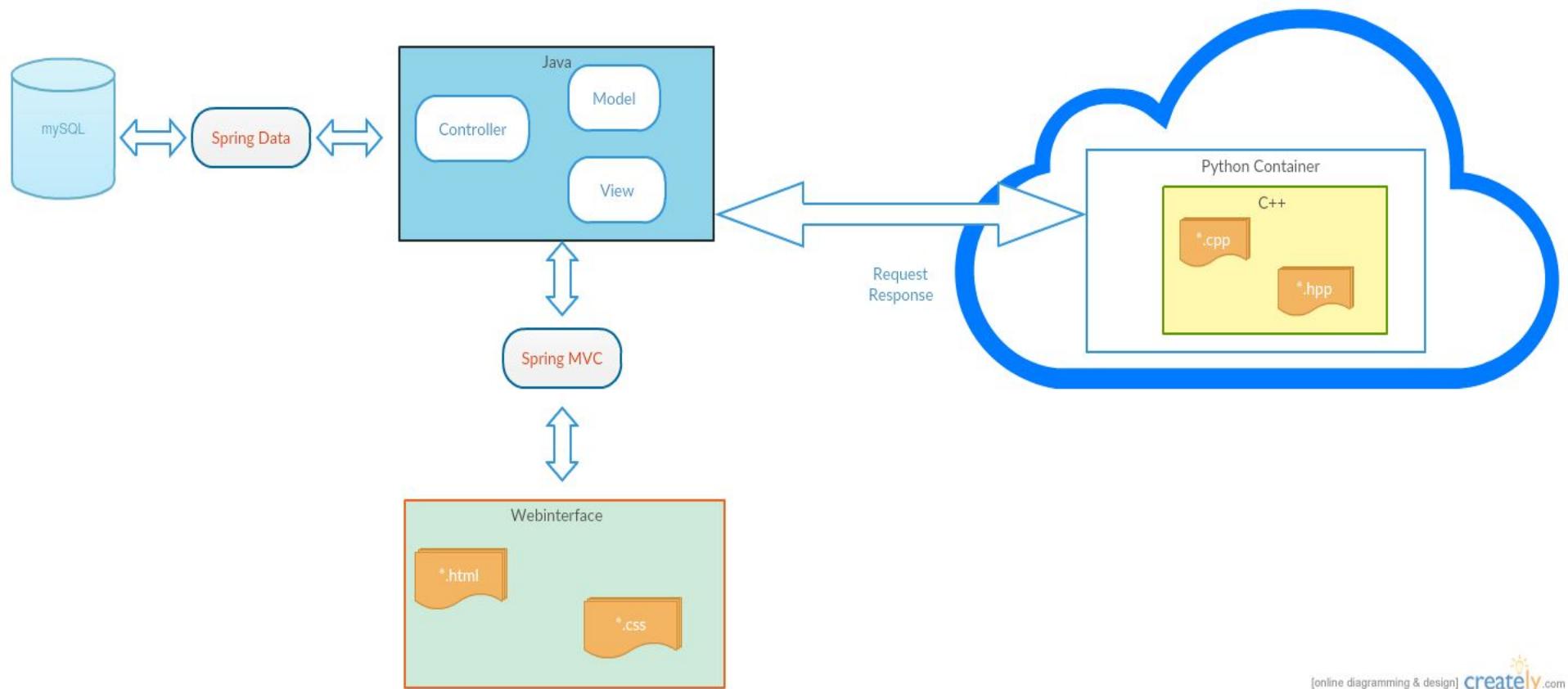
**statistics Table:**

- id BIGINT(20)
- ages TINYBLOB
- calculation\_time TINYBLOB
- is\_face INT(11)
- is\_no\_face INT(11)

**Indexes:** [button]

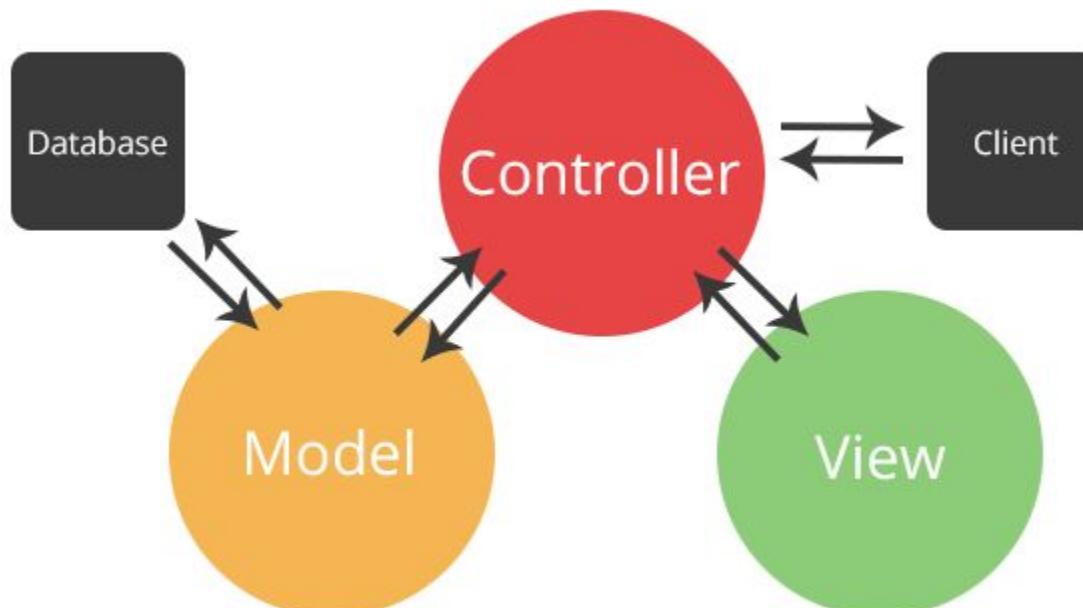
# sapaca

## MVC



# sapaca

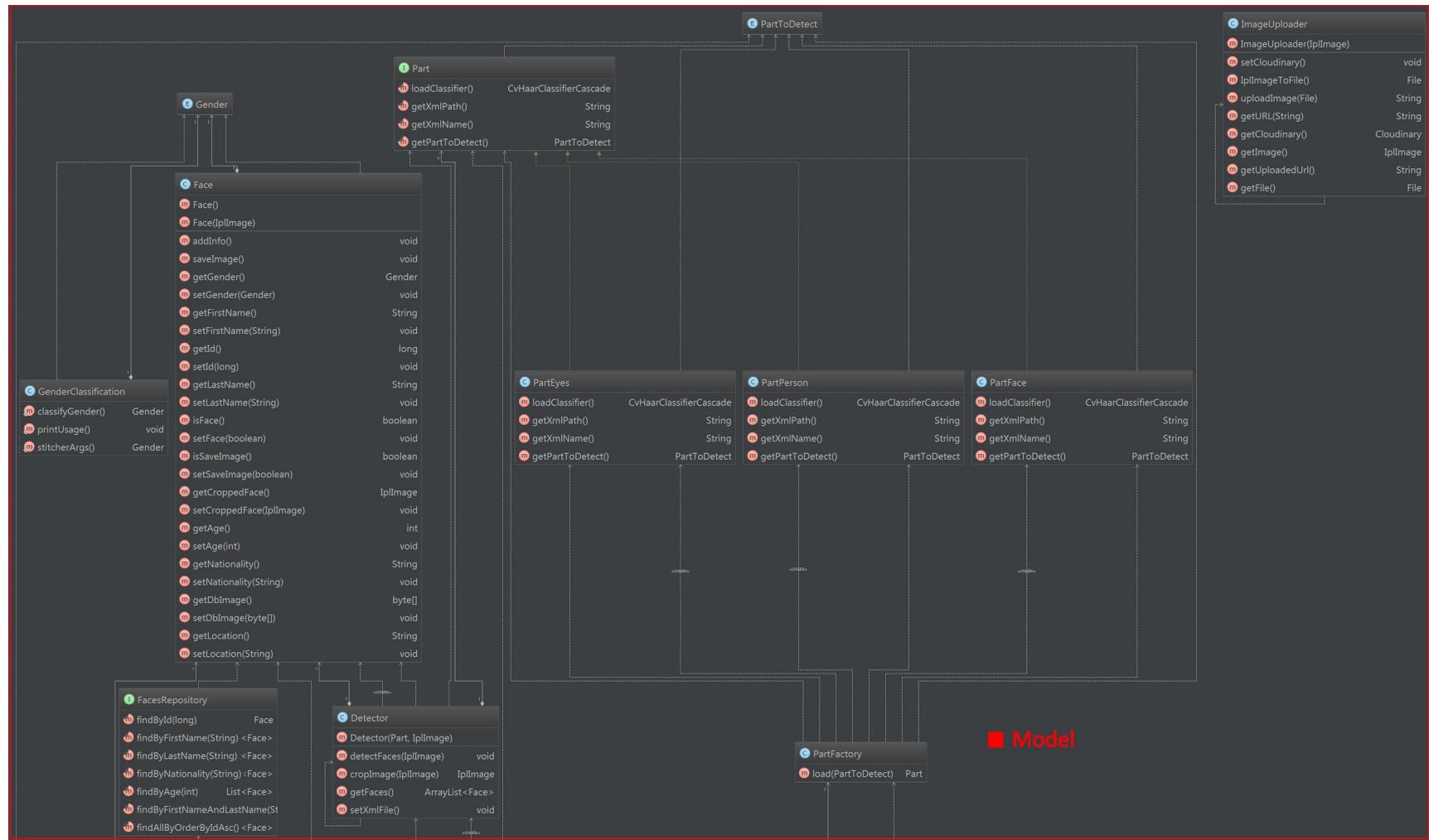
## MVC





# sapaca

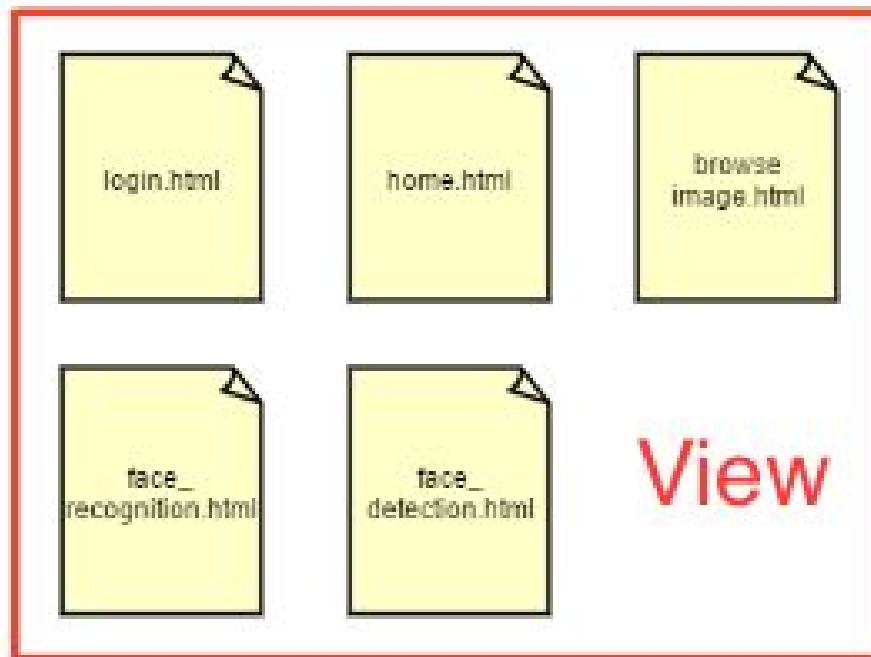
## MVC Model



■ Model

# sapaca

## MVC - View



# sapaca

## MVC - Controller

The screenshot shows a Java IDE interface with a class hierarchy diagram. On the left, there is a dark gray panel with a blue square icon and the word "Controller". To its right is a vertical dashed-line-based hierarchy diagram. At the top is a light gray box labeled "MainController". Below it is a list of methods, each preceded by a pink circle icon with a lowercase "m":

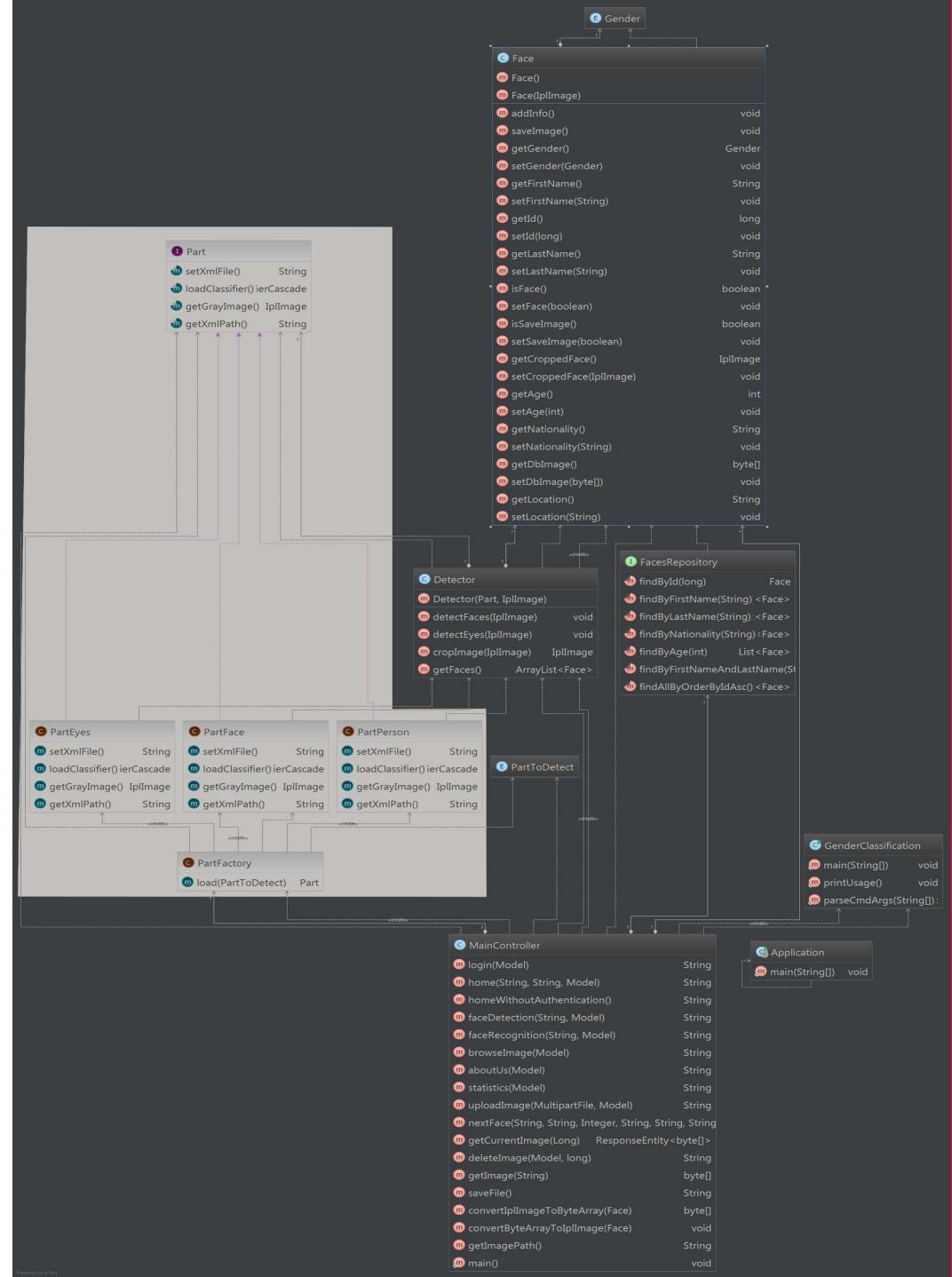
Method	Type
login()	String
home(String, String, Model)	String
homeWithoutAuthentication()	String
faceDetection()	String
faceRecognition()	String
browseImage(Model)	String
aboutUs()	String
statistics()	String
uploadImage(MultipartFile, Model)	String
nextFace(String, String, Integer, String, String)	String
getCurrentImage(Long) <byte[]>	byte[]
deleteImage(long)	String
getImage(String)	byte[]
saveFile()	String
convertIplImageToByteArray(Face)	byte[]
getImagePath()	String
test()	String

At the bottom of the hierarchy diagram is another light gray box labeled "Application". Inside this box, there is a method listed:

Method	Type
main(String[])	void

# sapaca

## Factory Pattern

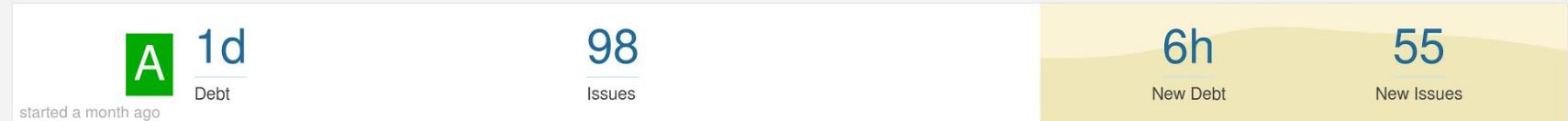


# QUALITY ASSURANCE

# sapaca

## SonarQube

Technical Debt [More >](#)



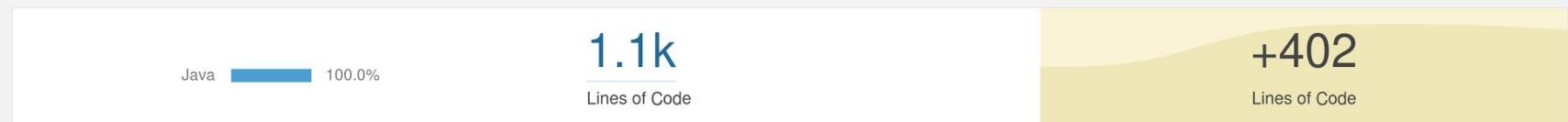
Coverage [More >](#)



Duplications [More >](#)



Structure [More >](#)



SonarQube™ technology is powered by SonarSource SA

Standard outputs should not be... ×

A field should not duplicate t... ×

Class variable fields should n... ×

Unused private fields should b... ×

Utility classes should not hav... ×

Instance method

# sapaca

## SonarQube

204 sudd... 205

```
@RequestMapping(value = "/next_face.html", method = RequestMethod.POST)
public String nextFace(@RequestParam(value = "firstName", required = false) String firstName,
```

The Cyclomatic Complexity of this method "nextFace" is 18 which is greater than 10 authorized. [...](#) a month ago ▾ L205 🔍 ⚡

Major Open Not assigned Not planned 18min debt brain-overload

206
207
208
209 naga...
210 sudd...
211
212 naga...
213 naga...
214 naga...
215
216 sudd...
217 naga...
218 naga...
219 naga...
220 naga...
221 naga...

```
@RequestParam(value = "lastName", required = false) String lastName,
@RequestParam(value = "age", required = false) Integer age,
@RequestParam(value = "nationality", required = false) String nationality,
@RequestParam(value = "gender", required = false) String gender,
@RequestParam(value = "location", required = false) String location,
@RequestParam(value = "faceDetected", required = false) String faceDetected,
@RequestParam(value = "noFaceDetected", required = false) String noFaceDetected,
@RequestParam(value = "genderClassification", required = false) String genderClassification,
@RequestParam(value = "eyeDetection", required = false) String eyeDetection,
@RequestParam(value = "attributes", required = false) String attributes, RedirectAttributes model) {
```

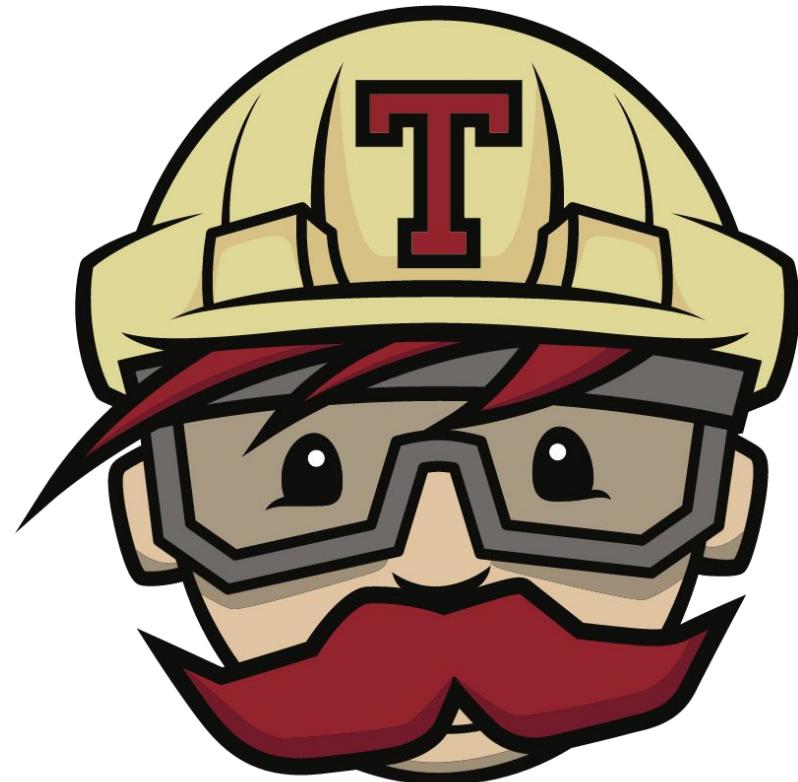
```
if (genderClassification != null) {
    Gender classifiedGender = new GenderClassification(faces.get(0).getCroppedFace()).getGender();
    model.addFlashAttribute("is_face_detected", "true");
    model.addFlashAttribute("classified_gender", classifiedGender.toString());
}
```

# sapaca

## Travis CI

- Automatic Deployment
- Build
- Test
- JIRA Task im Falle von einem Fail

build passing





Search all repositories



## My Repositories +

✓ sapacaFaceRecognition/FaceRec # 167

⌚ Duration: 5 min 27 sec

📅 Finished: about 13 hours ago

## sapacaFaceRecognition / FaceRecognition build passing

Current Branches Build History Pull Requests

✓ master changed xml file method;

⌚ #167 passed

Commit 31601ba

⌚ Elapsed time 5 min 27 sec

Compare 456de5a..31601ba

📅 about 13 hours ago

SuddenH4X authored and committed

```
1 Using worker: worker-linux-docker-b4cf114c.prod.travis-ci.org:travis-linux-6
2
3 ▶ Build system information
67
68 $ export DEBIAN_FRONTEND=noninteractive
69 $ git clone --depth=50 --branch=master https://github.com/sapacaFaceRecognition/FaceRecognition.git sapacaFaceRecognition/FaceRecognition
70 $ sudo service mysql start
71
72 This job is running on container-based infrastructure, which does not allow use of 'sudo', setuid and setgid executables.
73 If you require sudo, add 'sudo: required' to your .travis.yml
74 See https://docs.travis-ci.com/user/workers/container-based-infrastructure/ for details.
75
76 Setting environment variables from repository settings
77 $ export GH_TOKEN=[secure]
78 /home/travis/build.sh: line 45: export: `13b5064a8b5cc16ef4e540c2d4dc919c0152779': not a valid identifier
79 $ export jira_user=Ratnar
80 $ export jira_pass=[secure]
81
```

# sapaca

## Travis CI



sapaca vision / SAVI-102

Travis Build Error: master

Bearbeiten Kommentar Zuweisen Weitere Aktionen ▾

Aufgaben In Arbeit Fertig

### Details

Typ:	<input checked="" type="checkbox"/> Bug	Status:	<b>AUFGABEN</b> (Arbeitsablauf anzeigen)
Priorität:	Medium	Lösung:	Nicht erledigt
Stichwörter:	Keine		

### Beschreibung

The build process of commit: 849f35bf375bb29c674f0b5f2baff8117752492f was not successful. Please visit <https://travis-ci.org/sapacaFaceRecognition/FaceRecognition/builds/137612153> This information was automatically created. Please add further instructions.

### Aktivität

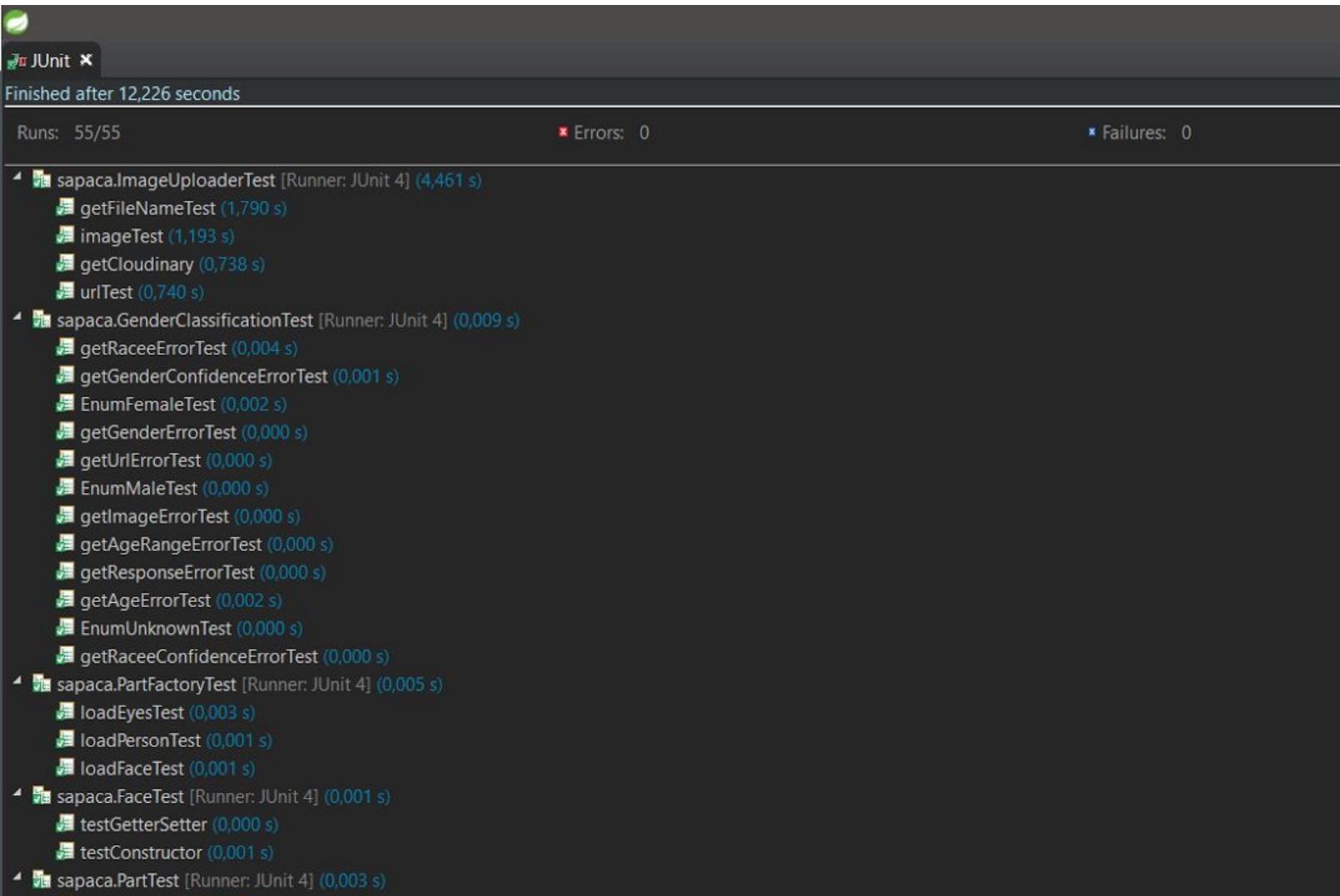
Alle **Kommentare** Arbeitsprotokoll Änderungshistorie Aktivität

Es liegen noch keine Kommentare zu diesem Vorgang vor.

Kommentar

# sapaca

## JUnit Testing



JUnit

Finished after 12,226 seconds

Runs: 55/55 Errors: 0 Failures: 0

- sapaca.ImageUploaderTest [Runner: JUnit 4] (4,461 s)
  - getFileNameTest (1,790 s)
  - imageTest (1,193 s)
  - getCloudinary (0,738 s)
  - urlTest (0,740 s)
- sapaca.GenderClassificationTest [Runner: JUnit 4] (0,009 s)
  - getRaceeErrorTest (0,004 s)
  - getGenderConfidenceErrorTest (0,001 s)
  - EnumFemaleTest (0,002 s)
  - getGenderErrorTest (0,000 s)
  - getUrlErrorTest (0,000 s)
  - EnumMaleTest (0,000 s)
  - getImageErrorTest (0,000 s)
  - getAgeRangeErrorTest (0,000 s)
  - getResponseErrorTest (0,000 s)
  - getAgeErrorTest (0,002 s)
  - EnumUnknownTest (0,000 s)
  - getRaceeConfidenceErrorTest (0,000 s)
- sapaca.PartFactoryTest [Runner: JUnit 4] (0,005 s)
  - loadEyesTest (0,003 s)
  - loadPersonTest (0,001 s)
  - loadFaceTest (0,001 s)
- sapaca.FaceTest [Runner: JUnit 4] (0,001 s)
  - testGetterSetter (0,000 s)
  - testConstructor (0,001 s)
- sapaca.PartTest [Runner: JUnit 4] (0,003 s)

# sapaca

## Code Coverage

coverage 41%

# SAPACAFACERECOGNITION / FACERECOGNITION

## 41%

BRANCH: MASTER ▾

NOTIFICATIONS

CHANGE SOURCE

GITHUB REPO

## LATEST BUILDS

BUILD	BRANCH	COVERAGE	COMMIT	COMMITTER	TYPE	TIME	VIA
#132	master	— 40.89	Configured build d.gradle file. It no	 caroem	push	28 May 2016	travis-ci



## LANGUAGES

JAVA, HTML, JAVASCRIPT, CSS, SHELL, BATCHFILE

REPO ADDED  
07 MAY 2016

TOTAL FILES  
13

BUILDS  
27

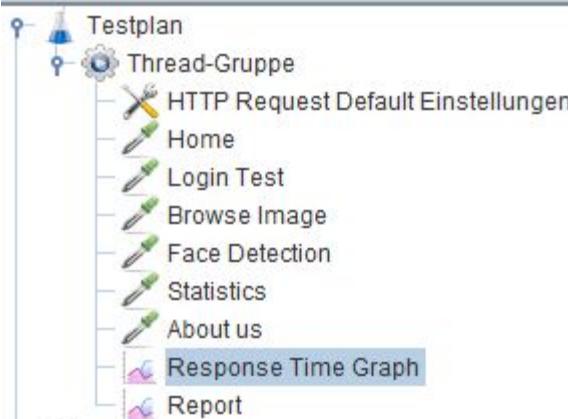
LAST BUILD  
#132 ↗

JOBS

LAST JOB

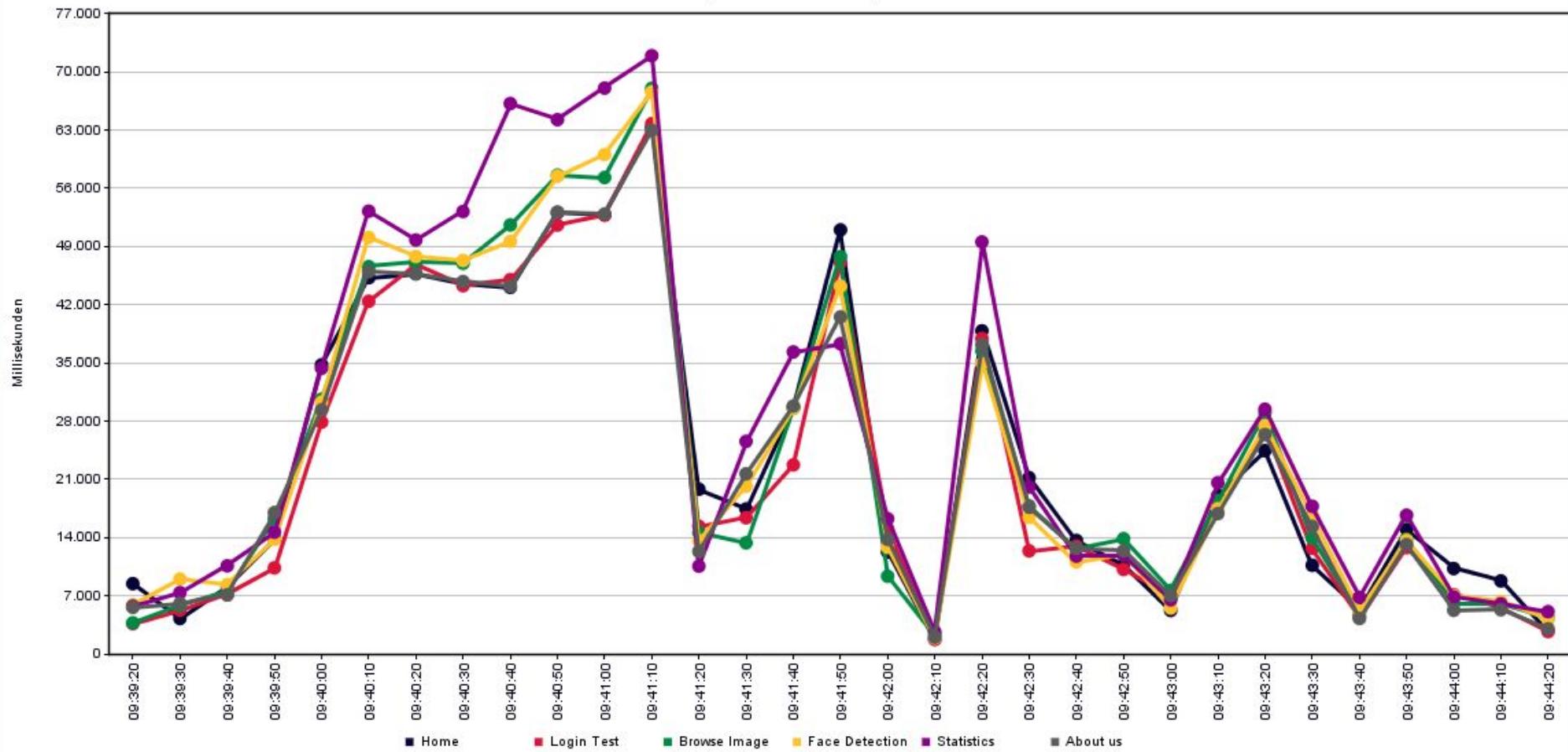
# sapaca

## Load Test



Settings Graph

Response Time Graph



**sapaca**

## Funktionale Tests

- Funktionale Tests wurde mit **cucumber** realisiert
- Automatisiertes Testen der Webanwendung wurde mit **Selenium** realisiert



# sapaca

## Feature File

```
④ Feature: Upload Image
  ④ Background: User is logged in
    ④ Scenario: Successful upload of an Image and FaceDetection
      Given I am logged in
      When I navigate to page "Face Detection"
      And I press the "Upload Image" Button
      And I select an Image with size "3" MB
      And I press the "Submit" Button
      Then I see the uploaded Image with detected Faces

    ④ Scenario: Not Successful upload of an Image
      Given I am logged in
      When I navigate to page "Face Detection"
      And I press the "Upload Image" Button
      And I select an Image with size "6" MB
      And I press the "Submit" Button
      Then I see an error message
```

# sapaca

## Feature File

Scenario: Not Successful upload of an Image  
Given I am logged in  
When I navigate to page "Face Detection"  
And I press the "Upload Image" Button  
And I select an Image with size "3" MB  
And I press the "Submit" Button  
Then I see the uploaded Image with detected Faces

```
# /Users/caro/Documents/P/Studium/DH/3.Ser
# Test_Steps.i_am_logged_in()
# Test_Steps.i_navigate_to_page(String)
# Test_Steps.i_press_the_button(String)
# Test_Steps.i_select_an_image(int)
# Test_Steps.i_press_the_button(String)
# Test_Steps.i_see_uploaded_image()
```

2 Scenarios (2 passed)  
12 Steps (12 passed)  
0m27.160s

---

# **sapaca**

## **Test Plan**

- Welche Tests wurden durchgeführt?
- Integration Workflow
- <https://github.com/sapacaFaceRecognition/Documentation/blob/master/Test%20Plan/Test%20Plan.pdf>

# sapaca

## Deployment Workflow



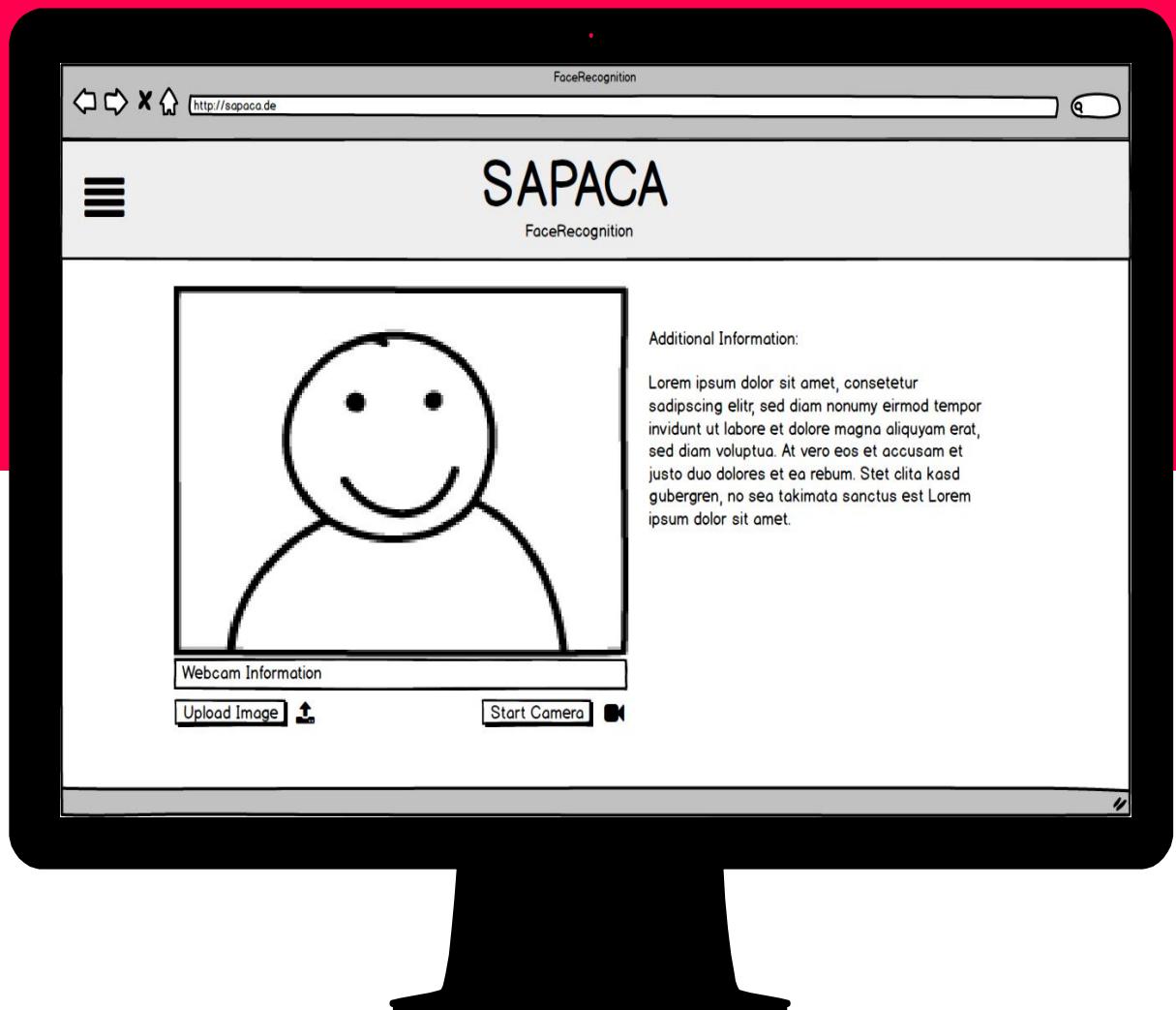
# sapaca

## Metrics

- Cyclomatic Complexity
- Response for Class
- Vorteile:
  - Kleinere zusammenhängende Codeabschnitte
  - Bessere Lesbarkeit
  - Bessere Wartbarkeit
  - Einfacheres Testen

# sapaca

## Demo



# Danke!



Gibt es noch offene Fragen?

**Blog:** <https://sapacablog.wordpress.com>

**GitHub:** <https://github.com/sapacaFaceRecognition/FaceRecognition>

**Doku:** <https://github.com/sapacaFaceRecognition/Documentation>

