



Hello,
World^{Inc}

B.R.A.D. DEMO 2

BY HELLO WORLD! INC

OVERVIEW

- Live Demo
- Quality Requirements
- Architectural Patterns
- Automated Testing
- Technology choices

DOCUMENTS LINK



LIVE DEMO



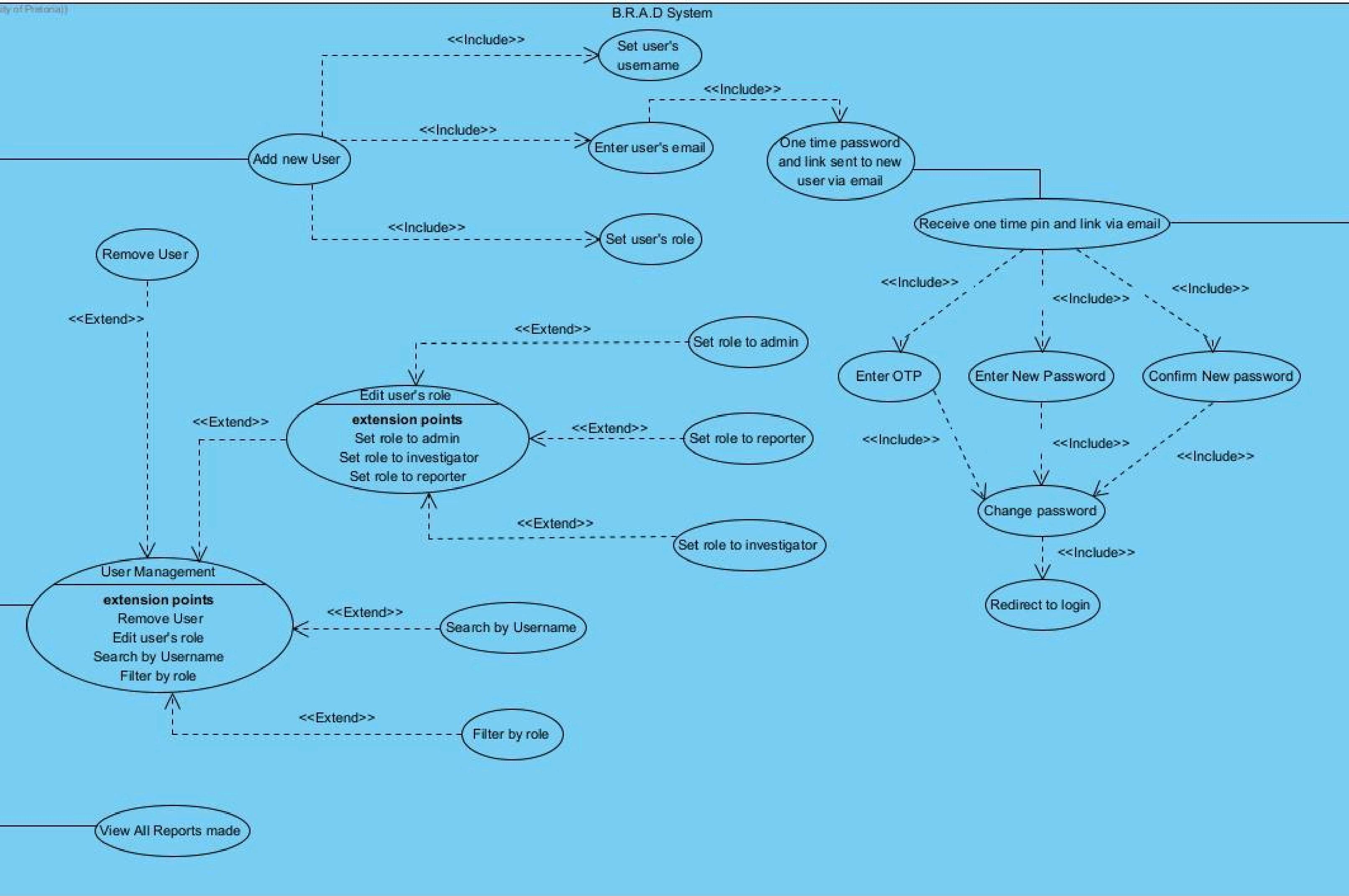
USECASES

USECASE 1 : ADMIN -ADD A NEW USER

- ADMIN USER IS ABLE TO
 - SET USER'S USERNAME
 - SET USER'S EMAIL
 - ONE TIME PASSWORD WILL BE EMAILED TO USER
 - USER MUST FOLLOW THE LINK IN EMAIL AND THEN:
 - ENTER OTP
 - ENTER NEW PASSWORD
 - CONFIRM NEW PASSWORD
 - THE REDIRECTED TO LOGIN AND ABLE TO LOGIN WITH NEW PASSWORD
 - SET USER'S ROLE

USECASE 2 : ADMIN'S USER MANAGEMENT

- ADMIN USER IS ABLE TO
 - REMOVE USER
 - EDIT USER'S ROLE
 - SET AS ADMIN
 - SET AS INVESTIGATOR
 - SET AS REPORTER
 - FILTER USERS BY ROLE
 - SEARCH USERS BY USERNAME

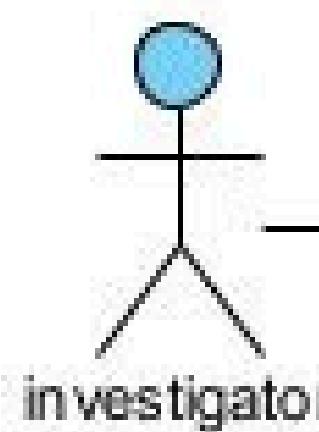


USECASES

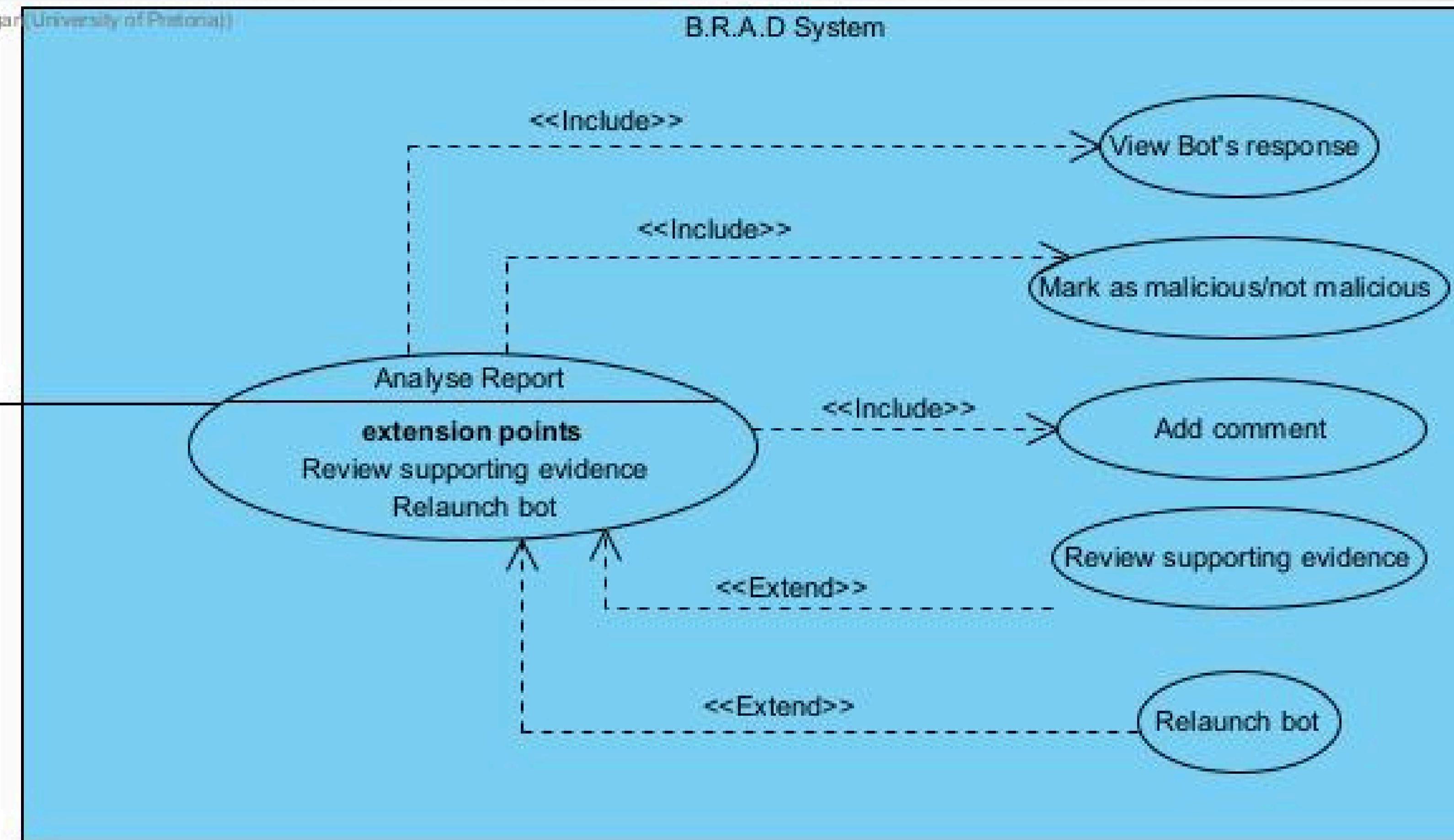
USE-CASE 3: INVESTIGATOR ANALYZING REPORT WITH BOT'S RESPONSE

THE INVESTIGATOR IS ABLE TO:

- ANALYZE A REPORT
- VIEW BOT'S RESPONSE
- MARK AS MALICIOUS/NOT MALICIOUS
- ADD COMMENT
- REVIEW SUPPORTING EVIDENCE OR RELAUNCH BOT



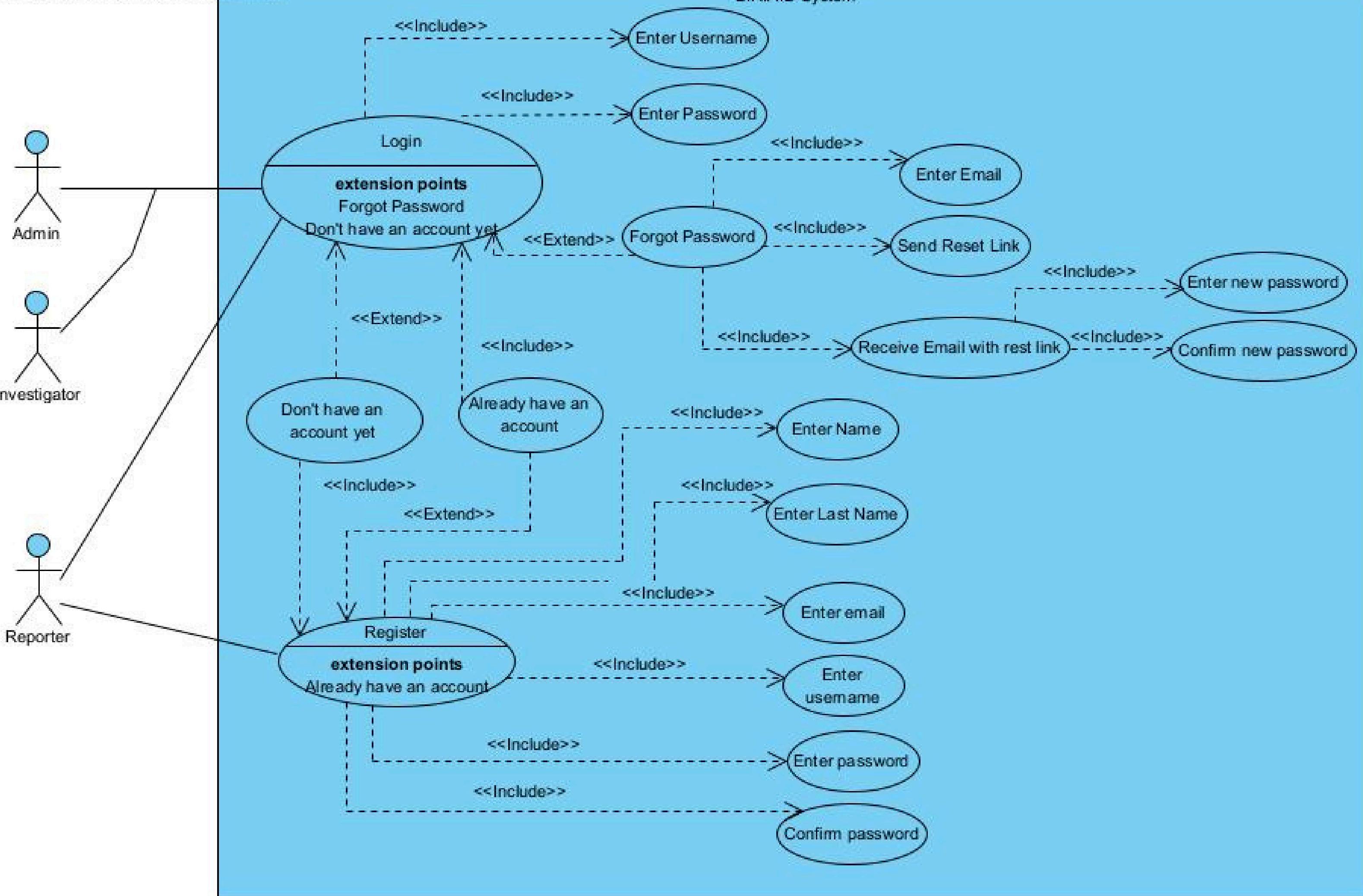
investigator



USECASES

USECASE 4: RESET PASSWORD

- ANY USER THAT HAS FORGOTTEN THEIR PASSWORD CAN REQUEST A LINK VIA EMAIL TO REST THEIR PASSWORD
- FORGOT PASSWORD
 - ENTER EMAIL
 - SEND REST LINK
 - RECEIVE EMAIL WITH RESET LINK
 - ENTER NEW PASSWORD
 - CONFIRM NEW PASSWORD



QUALITY REQUIREMENTS [RANKED]

1. SECURITY (MOST IMPORTANT)

100% of sensitive data encrypted at rest and in transit.

2. COMPLIANCE

Ensures that the system operates within the legal and ethical boundaries defined by regulations like GDPR and POPIA.

QUALITY REQUIREMENTS [RANKED]

3. RELIABILITY

Ensures that forensic investigations can be conducted consistently and accurately.

99.9% uptime, bot recovers from crashes within 60 seconds.

4. SCALABILITY

Supports 500+ concurrent domain submissions with average analysis < 10s/domain.

QUALITY REQUIREMENTS [RANKED]

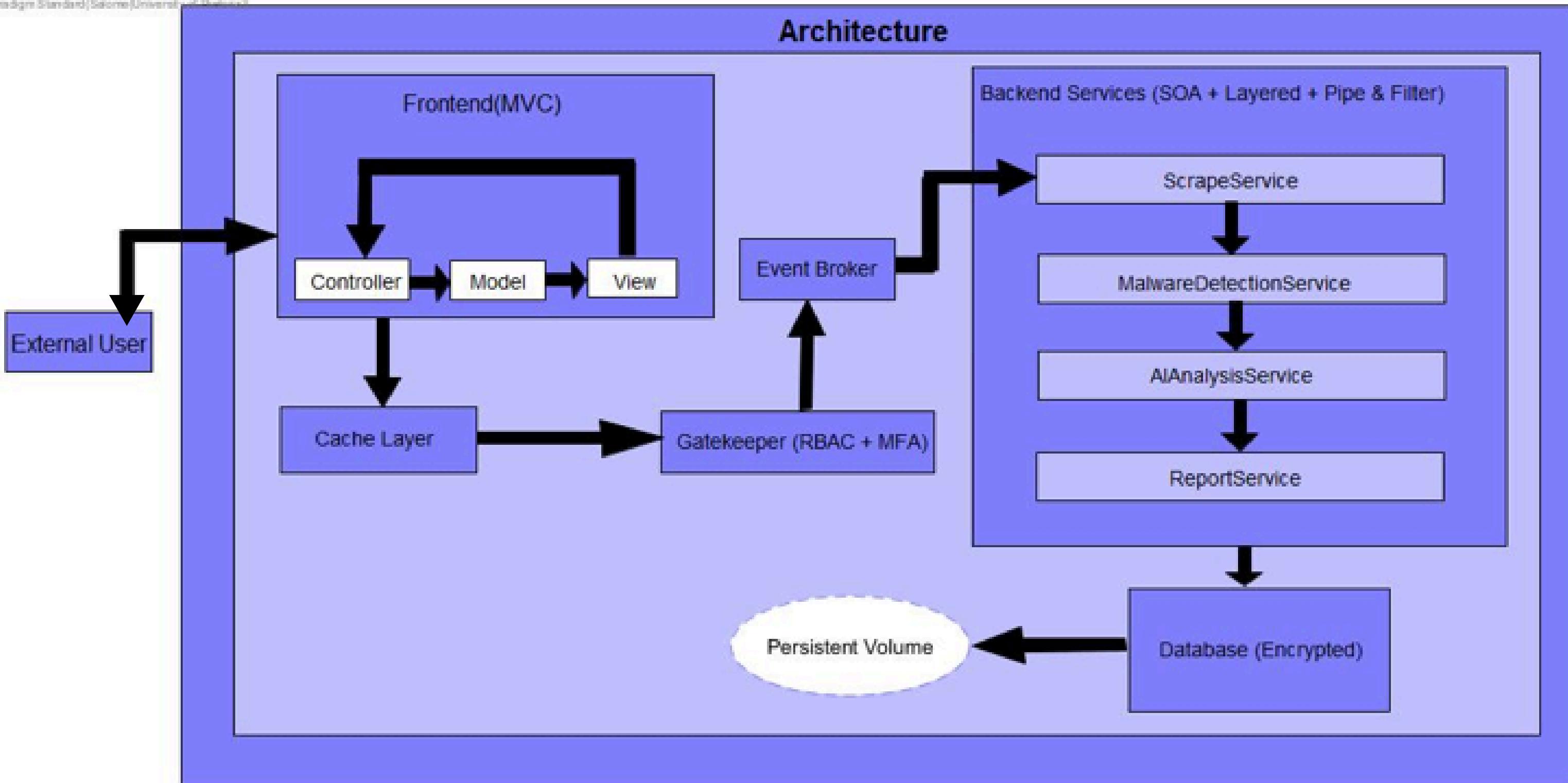
5. MAINTAINABILITY

Architecture must allow for frequent updates such as patching vulnerabilities, integrating new threat intelligence feeds or adapting AI models.

Docker-based components, automated deployment pipeline, <5 min rollout

ARCHITECTURE DIAGRAM

Visual Paradigm Standard Schema (Universal Class Diagram)



ARCHITECTURAL PATTERNS

01 Pipe and Filter Pattern

Quality Requirements Addressed:

- Maintainability: Filters can be updated without breaking the flow.
- Reliability: Pipeline resumes from failed stages where possible.
- Performance: Processing steps can be parallelized and optimized.

Scrape → Detect Malware → AI Risk Analysis → Metadata
Logging → Report Generation

ARCHITECTURAL PATTERNS

02 Gatekeeper Pattern

Quality Requirements Addressed:

- Security: All requests are authenticated, authorized, and validated before reaching internal services, preventing unauthorized access and injection attacks.
- Reliability: The Gatekeeper handles rate limiting, failover routing, and input filtering to protect internal services from overload or failure.
- Compliance: Every access attempt is logged and checked against regulatory rules, ensuring adherence to GDPR and POPIA obligations.

ARCHITECTURAL PATTERNS

03 Event-Driven Architecture (EDA)

Quality Requirements Addressed:

- Scalability: New event consumers can be added horizontally to meet increased demand during peak investigation periods.
- Performance: Asynchronous processing enables faster throughput for multiple domain investigations running in parallel.
- Reliability: Persistent queues and event logs allow retries and recovery if services fail mid-process.

ARCHITECTURAL PATTERNS

04 Service-Oriented Architecture (SOA)

Quality Requirements Addressed:

- Scalability: Each service can be scaled based on load without affecting the others.
- Maintainability: Services can be independently updated or replaced, supporting long-term evolution.
- Interoperability: Services follow standard formats and protocols (e.g., JSON, HTTP), enabling integration with external threat intel APIs.

AUTOMATED TESTING

```
● $ pytest --cov=src tests
=====
platform win32 -- Python 3.12.3, pytest-8.4.1, pluggy-1
rootdir: C:\Universiteit - code\2025 - COS 301\Capstone
plugins: cov-6.2.1
collected 23 items
```

```
tests\test_analysis.py ...
tests\test_bot.py .....
tests\test_metadata.py .....
tests\test_reporter.py .....
```

Name	Stmts	Miss	Cover
<hr/>			
src__init__.py	0	0	100%
src\bot.py	67	5	93%
src\utils__init__.py	0	0	100%
src\utils\analysis.py	46	2	96%
src\utils\metadata.py	56	0	100%
src\utils\reporter.py	23	0	100%
<hr/>			
TOTAL	192	7	96%
<hr/>			

```
PS C:\Users\megan\OneDrive\Desktop\Cos 301\Capstone\BRAD\BRAD

> frontend@0.0.0 test
> jest

PASS  src/_tests_/InvestigatorDashboard.test.jsx (13.453 s)
PASS  src/_tests_/Register.test.jsx (13.775 s)
PASS  src/_tests_/Landing.test.jsx (13.843 s)
PASS  src/_tests_/Login.test.jsx (13.753 s)
PASS  src/_tests_/ReporterDashboard.test.jsx (14.158 s)
PASS  src/_tests_/UserSettings.test.jsx (16.222 s)

● Console

    console.log
      Fields to update: { firstName: 'Alice' }

      at log (src/pages/UserSettings.jsx:46:13)

Test Suites: 6 passed, 6 total
Tests:       20 passed, 20 total
Snapshots:   0 total
Time:        22.794 s
Ran all test suites.

PS C:\Users\megan\OneDrive\Desktop\Cos 301\Capstone\BRAD\BRAD
```

```
TS auth.service.spec.ts M X  TS report.service.spec.ts M  TS report.controller.ts

api > src > auth > TS auth.service.spec.ts > ...
14  describe('AuthController (Unit)', () => {
30    it('should register a user', async () => {
41      expect(mockAuthService.register).toHaveBeenCalledWith(dto);
42    });
43  );
44);

it('should login a user', async () => {
  const dto: LoginDto = {
    identifier: 'janedoe',
    password: 'password123',
  };
  const result = await controller.login(dto);
  expect(result).toHaveProperty('token', 'mock-jwt-token');
  expect(mockAuthService.login).toHaveBeenCalledWith(dto);
});
});
```

```
EMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

at async Promise.all (index 1)
at TestingInstanceLoader.createInstances (node_modules/@nestjs/core/
at TestingInstanceLoader.createInstancesOfDependencies (node_modules/
at TestingInstanceLoader.createInstancesOfDependencies (node_modules/
at TestingModuleBuilder.createInstancesOfDependencies (node_modules/
at TestingModuleBuilder.compile (node_modules/@nestjs/testing/testin
at Object.<anonymous> (src/auth/auth.controller.spec.ts:8:35)

PASS  src/auth/auth.service.spec.ts (12.687 s)

Test Suites: 3 failed, 2 passed, 5 total
Tests:       2 failed, 4 passed, 6 total
Snapshots:   0 total
```

AUTOMATED TESTING

All workflows
Showing runs from all workflows

27 workflow runs

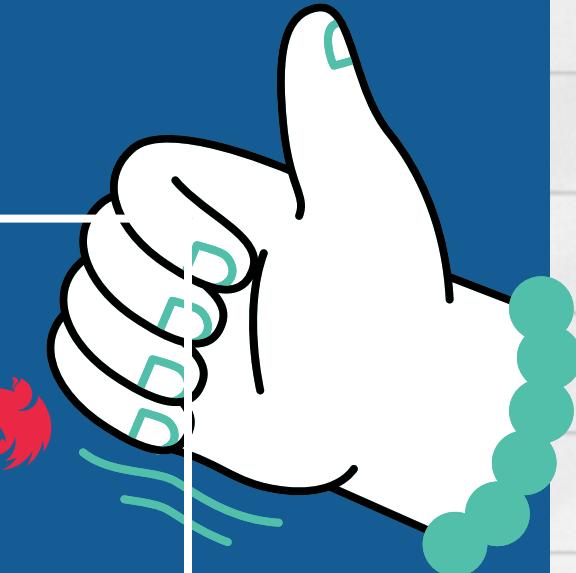
- ✗ Merge pull request #115 from COS301-SE-2025/apiUnitTesting
CI (No Docker) #7: Commit [2ed6212](#) pushed by Salome402 dev
- ✓ Merge pull request #115 from COS301-SE-2025/apiUnitTesting
CI/CD Pipeline #20: Commit [2ed6212](#) pushed by Salome402 dev
- ✓ Merge pull request #114 from COS301-SE-2025/ui
CI/CD Pipeline #19: Commit [7257f85](#) pushed by EthanVletter dev
- ✗ Merge pull request #114 from COS301-SE-2025/ui
CI (No Docker) #6: Commit [7257f85](#) pushed by EthanVletter dev
- ✓ Merge pull request #110 from COS301-SE-2025/ai-bot
CI/CD Pipeline #18: Commit [8663010](#) pushed by EthanVletter dev
- ✗ Merge pull request #110 from COS301-SE-2025/ai-bot
CI (No Docker) #5: Commit [8663010](#) pushed by EthanVletter dev
- ✗ Merge pull request #109 from COS301-SE-2025/docs
CI (No Docker) #4: Commit [73fd941](#) pushed by EthanVletter dev

CI/CD Pipeline

20 workflow runs

- ✓ Merge pull request #115 from COS301-SE-2025/apiUnitTesting
CI/CD Pipeline #20: Commit [2ed6212](#) pushed by Salome402 dev
- ✓ Merge pull request #114 from COS301-SE-2025/ui
CI/CD Pipeline #19: Commit [7257f85](#) pushed by EthanVletter dev
- ✓ Merge pull request #110 from COS301-SE-2025/ai-bot
CI/CD Pipeline #18: Commit [8663010](#) pushed by EthanVletter dev
- ✓ Merge pull request #109 from COS301-SE-2025/docs
CI/CD Pipeline #17: Commit [73fd941](#) pushed by EthanVletter dev
- ✓ Merge pull request #107 from COS301-SE-2025/ui
CI/CD Pipeline #16: Commit [58f570c](#) pushed by EthanVletter dev
- ✓ Merge pull request #105 from COS301-SE-2025/rolebyadmin
CI/CD Pipeline #15: Commit [957f97e](#) pushed by Megan-Pretorius dev
- ✓ Merge pull request #104 from COS301-SE-2025/docker-fix
CI (No Docker) #3: Commit [957f97e](#) pushed by EthanVletter dev

TECHNOLOGY CHOICES



- **Frontend**

- React.js 
- CSS 

- **Backend**

- NestJS (built on Node.js using  JavaScript)

- **Scraping Tools**

- BeautifulSoup (Python) 
- Playwright (Python) 

- **Programming languages**

- Javascript 
- TypeScript 
- CSS 
- Python 

- **DevOps & Collaboration Tools**

- Git 
- GitHub 
- Docker 
- Swagger 

**THANK
YOU VERY
MUCH!**